



TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA INFORMÁTICA  
MENCIÓN EN INGENIERÍA DE COMPUTADORES

## eHand: arquitectura de control para prótesis mio-eléctricas

**Estudiante:** Alejandro Iregui Valcárcel  
**Dirección:** Adriana Dapena Janeiro  
**Dirección:** Francisco Laport López

A Coruña, noviembre de 2021.



*Con mucho cariño, para el mundo.*



## **Agradecimientos**

Siempre agradecido de mi familia y amigos quiero, sobre todo, darle las gracias a mi madre por el apoyo y amor que me ha dado siempre.

A todos los usuarios que confiaron en mí para hacer las diferentes pruebas.

A mis directores igualmente, por creer en mí y ayudarme con este proyecto.

A Thais Pousada por sus consejos.

A la Cátedra UDC-everis por su apoyo económico para comprar algunos de los componentes hardware.



## **Resumen**

eHand es un proyecto tecnológico, social y totalmente Open Source en el que se busca integrar hardware y software para el análisis de señales musculares. Mediante sensores y electrodos, se medirán los pocos milivoltios que se generan durante la actividad muscular. Dichas señales eléctricas serán convertidas de analógico a digital por un Arduino. La información que se genere será enviada a un ordenador vía puerto serie. Finalmente, mediante herramientas de procesado de señales, se extraerán patrones de clasificación sobre los diferentes movimientos deseados para con ellos controlar cualquier tipo de interfaz hombre-máquina. Se contempla el control de juegos y un modelo 3D de movimiento de la mano.

## **Abstract**

eHand is a technological, social and totally Open Source project that seeks to integrate hardware and software for the analysis of muscle signals. Using sensors and electrodes, the few millivolts that are generated during muscle activity will be measured. This electrical signals are converted from analog to digital by using an Arduino board. The data will be sent to a computer via serial port. Finally, using tools for signal processing, an application obtains a pattern classification of the different desired movements in order to control any type of human-machine interface with them. It currently supports control of games and a 3D model of a hand.

### **Palabras clave:**

Electromiografía

Microcontrolador

Procesamiento de Señal

Máquina de vectores de soporte

Juegos

Modelo 3D

### **Keywords:**

Electromyography

Microcontroller

Signal processing

Support Vector Machine

Games

3D Model



# Índice general

---

<b>1. Introducción</b>	<b>1</b>
1.1. Motivaciones . . . . .	1
1.2. Prótesis mio-eléctricas . . . . .	2
1.3. Objetivos . . . . .	5
1.4. Organización de la memoria . . . . .	6
<b>2. Estado del Arte</b>	<b>7</b>
2.1. Anatomía básica . . . . .	7
2.2. Revisión del Hardware . . . . .	9
2.2.1. Electrodos . . . . .	9
2.2.2. Placas EMG . . . . .	10
2.3. Desarrollos conocidos . . . . .	16
2.4. Conclusiones . . . . .	20
<b>3. Arquitectura Desarrollada</b>	<b>21</b>
3.1. Tecnologías . . . . .	21
3.1.1. Arduino . . . . .	22
3.1.2. Python . . . . .	22
3.1.3. Blender . . . . .	24
3.2. Adquisición de señales . . . . .	24
3.2.1. Pruebas preliminares . . . . .	26
3.3. Procesado de señales . . . . .	28
3.3.1. Extracción de características . . . . .	28
3.3.2. Clasificación basada en umbral . . . . .	29
3.3.3. Clasificación utilizando máquina de vectores de soporte . . . . .	30
3.3.4. Pruebas preliminares . . . . .	31
3.4. Aplicaciones . . . . .	33
3.4.1. Videojuego de saltos . . . . .	34

3.4.2. Modelo prótesis 3D . . . . .	35
3.5. Código de acceso libre . . . . .	36
3.6. Conclusiones . . . . .	37
<b>4. Pruebas con usuarios</b>	<b>39</b>
4.1. Sujetos de prueba . . . . .	39
4.2. Protocolo . . . . .	40
4.3. Juego de saltos . . . . .	43
4.4. Modelo 3D . . . . .	46
4.5. Conclusiones . . . . .	49
<b>5. Metodología, planificación y costes del desarrollo</b>	<b>51</b>
5.1. Metodología . . . . .	51
5.1.1. Fase 1 y 2 . . . . .	52
5.1.2. Fase 3 y 4 . . . . .	52
5.1.3. Fase 5 y 6 . . . . .	52
5.2. Planificación . . . . .	52
5.3. Incidencias . . . . .	53
5.4. Estimación de costes . . . . .	55
<b>6. Conclusiones y líneas futuras</b>	<b>57</b>
6.1. Conclusiones . . . . .	57
6.2. Líneas futuras . . . . .	58
<b>Bibliografía</b>	<b>63</b>

# Índice de figuras

---

1.1.	Prótesis para niños de Open Bionics, Hero Arm [1]. . . . .	3
1.2.	Esquema de una protésis mio-eléctrica [2]. . . . .	4
2.1.	Potencial de acción en la membrana muscular [3]. . . . .	8
2.2.	Colocación de los electrodos durante los experimentos [4]. . . . .	10
2.3.	Esquemático de la placa Olimex [5]. . . . .	11
2.4.	Sensor EMG de Olimex [5]. . . . .	12
2.5.	Sensor EMG de origen chino [6]. . . . .	13
2.6.	Esquema del montaje para la placa china [6]. . . . .	14
2.7.	Sensor Myoware de 1 canal [7]. . . . .	15
2.8.	Sensor Gravity de 1 canal [8]. . . . .	15
2.9.	Sensor de 8 canales, Cyton de Open BCI [9]. . . . .	16
2.10.	EMG durante la sentadilla [10]. . . . .	17
2.11.	Activación de trapecio en paciente sano versus lesionado [10]. . . . .	18
3.1.	Esquema de la arquitectura. . . . .	21
3.2.	Montajes de las placas: Olimex (izquierda), placa de Aliexpress (derecha). . . . .	25
3.3.	Salida del Serial Plotter en el IDE Arduino para (a) extensión, (b) flexión, (c) reposo y (d) cierre de puño. . . . .	27
3.4.	RMS calculado sobre las mediciones hechas para la técnica de umbral: reposo (azul), flexión (naranja) y linea roja el umbral calculado. . . . .	31
3.5.	RMS para cada uno de los cuatro movimientos: reposo (azul), extensión (amarillo), cierre de puño (negro) y flexión (naranja). . . . .	32
3.6.	Mapa de clasificación para distintos modelos. . . . .	33
3.7.	Interfaz de la aplicación. . . . .	34
3.8.	Ejemplo de EMG como interfaz para juegos. . . . .	35
3.9.	Ejemplo de cierre de puño en el modelo 3D desarrollado. . . . .	36

4.1.	Mesotipos del cuerpo humano. . . . .	40
4.2.	Colocación de los electrodos secos Olimex. Extensor en la izquierda, flexor en la derecha. . . . .	41
4.3.	Esquema de la pauta seguida en las mediciones para la técnica del umbral. . .	43
4.4.	Usuario probando el juego. . . . .	44
4.5.	Resultados RMS (en orden) para los distintos usuarios: reposo (azul), flexión (naranja), linea roja el umbral calculado. . . . .	45
4.6.	Usuario probando el modelo 3D. . . . .	46
4.7.	Mapas de clasificación para los usuarios 1 y 2. . . . .	47
4.8.	Ejemplo de todos los movimientos conseguidos en el modelo 3D. . . . .	48
5.1.	Diagrama de Gantt de la planificación inicial del proyecto. . . . .	53
6.1.	Microcontrolador ESP32, con tecnología Wifi y Bluetooth. . . . .	60

# Índice de tablas

---

2.1.	Electrodos utilizados en este proyecto. . . . .	10
4.1.	Información de los sujetos para las pruebas. . . . .	40
4.2.	Umbral y número de intentos necesarios para superar un nivel del juego por cada usuario. . . . .	44
4.3.	Resultados para 3-fold en los 4 kernels y por cada usuario. . . . .	49
5.1.	Planificación inicial para el proyecto. . . . .	53
5.2.	Resumen temporal del proyecto. . . . .	54
5.3.	Resumen de costes del proyecto. . . . .	55
5.4.	Costes de recursos humanos. . . . .	55



## **Capítulo 1**

# **Introducción**

---

**E**l uso de prótesis por la falta de miembros en el cuerpo es una historia que nos persigue desde antes de Cristo. Debido a las limitaciones de la época, se desarrollaban implantaciones fijas cuya utilidad era muy limitada o ninguna. A lo largo de los años y con ayuda de los avances en electrónica y mecánica, el ser humano ha sido hábil para mejorarlas. Actualmente existen prótesis avanzadas que son capaces de devolver a la persona una importante variedad de movimientos. Aun así, debido a factores como el mercado de producción o la tecnología, nos encontramos con productos que no están a disposición de todo el mundo. Uno de los factores que limitan su uso es la relación precio/utilidad.

### **1.1. Motivaciones**

Este proyecto tiene como motivación principal avanzar en el desarrollo de una prótesis a bajo coste. El precio actual de una prótesis comercial ronda los 60.000€, algo que es inacequible para muchas personas y, sobre todo, en casos de pacientes muy jóvenes que necesitan varios cambios de prótesis a lo largo de su crecimiento. Se trata de un objetivo a medio-largo plazo que excede la realización del presente Trabajo Fin de Grado (TFG), pero que estamos seguros conseguiremos en un futuro. En este trabajo se abordan algunos de los pasos a seguir. El camino para conseguir los primeros logros no ha sido directo. Durante el estudio y realización del mismo surgieron problemáticas que tuvieron que ser resueltas y que comentaremos en esta memoria.

En cuanto a la facilidad que existe para acceder a herramientas de desarrollo en este tipo de trabajos en concreto, ya sean elementos hardware de sensorización, conversión Analógico-Digital y software especializado, fue difícil encontrar una plataforma con la que poder trabajar fácilmente y por poco precio. Por eso, en este TFG se hace un estudio de los elementos con el objetivo de ayudar a identificar las mejores opciones y ofrecer instrumentos de tipo Open Source que puedan ser utilizados por cualquier persona interesada en el tema. Sin embargo,

de poco sirve el desarrollo de herramientas si no existe un fácil acceso a documentación o divulgación sobre el tema. Desde la perspectiva de un estudiante en ingeniería, fue difícil encontrar información específica sobre Electromiografía (EMG) que sea fácil de digerir. Es por ello que otra motivación de este TFG es ofrecer los detalles necesarios para comprender cómo las señales EMG pueden ser transformadas en órdenes para el movimiento de una prótesis o cualquier otra máquina. Es decir, pretendemos explicar todos los pasos a seguir y ayudar a su divulgación.

Para llegar a conseguir que las prótesis electromecánicas sean asequibles y viables para convertirse en un producto, es interesante también analizar otros usos que se le podría dar, y que se le está dando, a esta tecnología. Cuanto más utilizada sea, mayor mercado se creará, menores serán los costes, existirán más desarrollos, mejor será la competencia y, en definitiva, los productos verán aumentada su calidad y accesibilidad. Esto nos ha llevado a no centrar el desarrollo solo en la prótesis, sino que también se ha estudiado cómo el producto final de este TFG puede ser usado para rehabilitación e, incluso, para entretenimiento.

En un sistema como este, que estará integrado en el cuerpo de una persona, existirán ciertos requisitos físicos como el tamaño de la implantación y peso que serán un obstáculo para muchas arquitecturas hardware. Será entonces necesario estudiarlas para cubrir las necesidades de los pacientes y concretamente, en el procesamiento digital de señales. Es muy común actualmente ver soluciones específicas o diseños a medida que cumplan con requisitos de bajo consumo, alta capacidad de procesamiento e incluso conectividad.

Con respecto a esto último y motivado hacia el estudio del codiseño hardware y software del sistema, es imposible no ver el auge actual del aprendizaje máquina como herramienta para el procesado de señales y lo bonito que resulta descubrir que se trata de una solución completamente natural y útil para clasificar movimientos mediante señales musculares.

Finalmente, existe la voluntad personal del estudiante en crear una comunidad de software Open Source para el estudio de este tipo de señales, con la intención de proporcionar lo necesario para permitir cualquier tipo de trabajo relacionado con el EMG, siempre bajo el lema de “vender hardware, no software”.

## **1.2. Prótesis mio-eléctricas**

En la actualidad, las prótesis más empleadas y simples de diseñar son tipo gancho (solo hacen la apertura y cierre de la mano). Durante esta sección queremos exponer la estructura general para su control mediante señales EMG. Su completo desarrollo excede a los objetivos de este TFG, pero el hecho de conocer sus necesidades ayudará a comprender las partes abordadas en nuestro desarrollo.

El hardware de una prótesis debe ser capaz de realizar lo siguiente:

- Leer señales EMG.
- Conversión Analógica-Digital.
- Procesar estas señales en forma de órdenes.
- Controlar el estado de los elementos mecánicos como motores y responder con la acción.
- Tener un buen firmware para la verificación del estado de componentes mecánicos, batería, actualizaciones, etc.



Figura 1.1: Prótesis para niños de Open Bionics, Hero Arm [1].

En primer lugar, es muy importante determinar el conjunto de componentes que se van a utilizar para cada una de estas necesidades. En un mundo perfecto deberíamos ser capaces de emplear la misma configuración para cualquier prótesis, pero, en realidad, dependeremos directamente del tipo de amputación o del paciente. Por ejemplo, si queremos diseñar una prótesis de mano para un niño como la que se muestra en la Figura 1.1, necesitaremos cumplir unos requisitos de tamaño y peso muy específicos. Conseguir esto sin sacrificar ciertos aspectos como la autonomía, la cantidad de componentes o la potencia de computación, es un trabajo notable. Por ello, sería interesante encontrar un equilibrio que permitiese producir diferentes prótesis sin necesidad de reinventarse en cada momento.

El software es también de una parte importante para un sistema como este. Está claro que para el correcto diseño dependeríamos de la configuración de componentes que se elijan. Pero, si se trabajase en una serie de herramientas que agilizasen el proceso de producción, estaríamos dando un gran paso. Creemos por ello que existe la necesidad de producir software libre, el cual ahorraría horas de trabajo y pruebas. La colaboración de una comunidad empujaría a

los desarrollos a alcanzar una escalabilidad que no se consigue mediante soluciones individuales. Creemos que actualmente estos son los mayores problemas a los que se enfrenta un sistema de este tipo. No existe una solución hardware y software lo suficientemente general que satisfaga estos requisitos y con la que puedan trabajar fácilmente los desarrolladores, sino que cada organización aporta una solución demasiado específica y nacida desde cero, lo que implica un tiempo y coste de desarrollo muy alto.

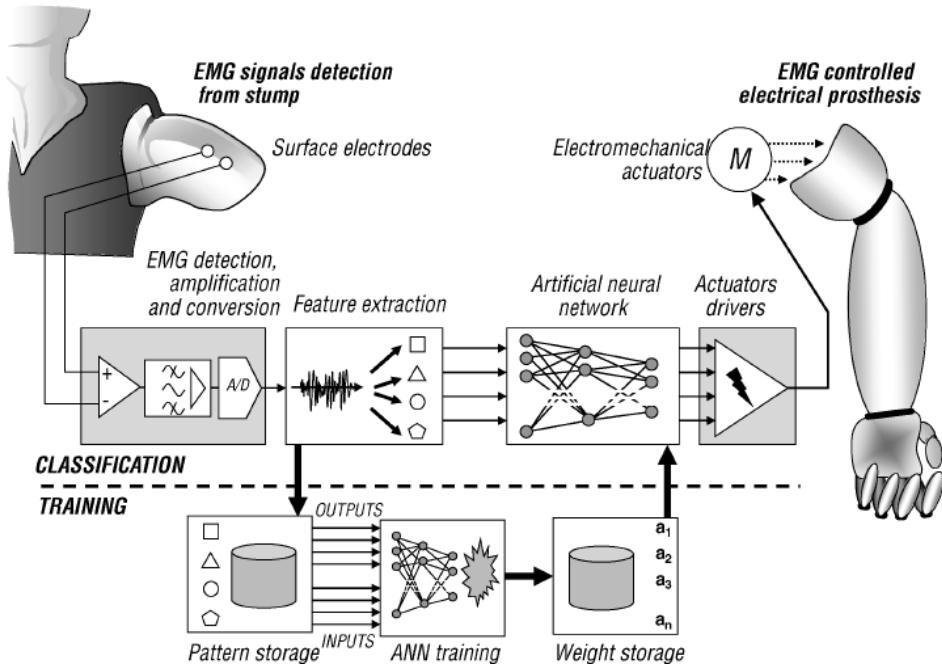


Figura 1.2: Esquema de una protésis mio-eléctrica [2].

Haciendo un ejercicio de análisis sobre la Figura 1.2, en cada etapa se podrían definir ciertos requisitos que podrían dar soluciones más generales y aplicables en diferentes casos.

En la primera etapa de adquisición de señales es normal que la calidad en la circuitería varíe por cada fabricante, pero se podría definir un cierto estándar de calidad y características en la señal analógica que generan. Por ello en el capítulo 2 revisaremos varios sensores.

Veremos durante la revisión que estos sensores comparten características en la señal analógica que producen por la compatibilidad que buscan tener sobre placas de desarrollo tipo Arduino. Es decir, este tipo de microcontrolador de una manera u otra está obligando a los fabricantes a seguir una misma tendencia en cuanto a los productos sensoriales que ofrecen. Con respecto al número de sensores, se podría definir un equilibrio entre el tipo de amputación y cantidad de movimientos a reconocer. La pierna por ejemplo, tiene menos movimientos que una mano, por lo que no sería lógico utilizar la misma cantidad de sensores en todos los ca-

sos. Igualmente, este análisis también dependerá directamente de que las siguientes fases de extracción de características y clasificación sean capaces de sacarles provecho.

Si contamos con una cierta estandarización en la adquisición de señales, las siguientes 2 etapas se podrían desarrollar de forma generalizada y Open Source. En la extensa bibliografía consultada, cada autor presenta soluciones y configuraciones diferentes. Si cada investigador se va a dedicar a reinventar la rueda todo el rato, muchos de estos desarrollos probablemente se queden sin aplicar u olvidados. Se podría trabajar de forma comunitaria en un diseño software que se dedicase a solucionar el problema de extraer la información necesaria para mimetizar el miembro humano en base a unas condiciones de sensorización.

También es importante que en la arquitectura se haga una división de hardware, del procesado de señales y del resto de componentes. Es decir, lo que le da capacidad al sistema para percibir el mundo real y actuar de forma inteligente, debería estar separado del comportamiento que comúnmente conocemos más secuencial. Lo que buscamos es querer darle flexibilidad y generalidad al sistema para que sea capaz de albergar fácilmente, alguno de los métodos de procesado software que se hayan desarrollado para esta causa.

Comprendidas las fases más importantes en la producción de una prótesis, dejemos a los fabricantes dedicar sus esfuerzos a la producción y venta de más bajo nivel. Lo interesante es que cada vendedor defina la relación calidad/precio que le quiere dar a su producto, para ofrecer diferentes alternativas al mercado.

### 1.3. Objetivos

El objetivo final, a medio-largo plazo, es tener una prótesis de un brazo-mano como la descrita en la sección anterior. Sin embargo, esto requiere realizar un desarrollo tecnológico que está fuera del presente TFG. Es por ello, que en el presente proyecto se fijaron seis objetivos concretos:

- Hacer una revisión de la información más importante y estado actual de la tecnología usada en EMG.
- Encontrar y analizar el hardware para adquisición de señales que sea de bajo coste y útil para los experimentos.
- Construir herramientas software de tipo software libre para el tratamiento y clasificación de las señales.
- Utilizar estas herramientas para diferentes aplicaciones como un juego y el control de un modelo 3D.
- Hacer experimentos y probar las aplicaciones en usuarios reales.

- Documentar todo el proceso y destacar los aspectos más relevantes que se aprenden con la experiencia de trabajar con el equipamiento para que, en un futuro, cualquiera pueda replicar con facilidad lo que se hizo en este TFG.

## **1.4. Organización de la memoria**

Este documento pretende reflejar la experiencia del alumno en su paso por descubrir de cero lo que significa construir un sistema de este tipo. La memoria está organizada en forma de guía, donde se empezará introduciendo conceptos básicos: la biología que existe detrás de estas señales, la instrumentación para su adquisición, el procesado, y finalmente, los capítulos en donde se hablará de las aplicaciones desarrolladas y de los experimentos con usuarios.

- Capítulo 2: Estado del arte. Se comentará la situación actual de la tecnología EMG y se introducirán algunos de los conceptos biológicos en los que se basa.
- Capítulo 3: Arquitectura desarrollada. Se describirán los procedimientos y conceptos teóricos sobre las herramientas que se utilizan para procesar las señales utilizadas y se hablará de las aplicaciones finales.
- Capítulo 4: Pruebas con usuarios. Se presentarán y describirán los experimentos realizados.
- Capítulo 5: Metodología. Se describe la metodología empleada, la planificación y los costes de desarrollo.
- Capítulo 6: Conclusiones del trabajo y líneas futuras.

## Capítulo 2

# Estado del Arte

---

**E**n este capítulo se comentará la situación actual de la tecnología EMG tanto a nivel del hardware que se puede emplear como de aplicaciones que han sido realizadas por otros investigadores.

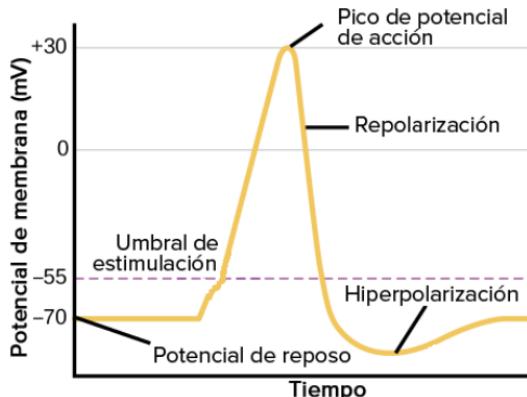
Todo análisis, sea cual sea el desarrollo, se basa principalmente en el concepto de que con la electromiografía, buscamos extraer información de la actividad muscular para reconocer patrones de movimientos y extraer características de cada uno de los músculos implicados. Es por ello que comenzaremos explicando algunos de los conceptos biológicos en los que se basa la tecnología EMG.

### 2.1. Anatomía básica

Los músculos son un tipo de tejido blando que puede contraerse mediante impulsos nerviosos, generando movimiento y permitiendo trabajos mecánicos. La electromiografía es una técnica que mide la actividad eléctrica generada por el paso del impulso nervioso, que provoca la despolarización en la membrana de la célula muscular durante la excitación. Si la despolarización alcanza un determinado valor umbral, se genera un potencial de acción [2, 3].

Por lo tanto, la EMG mide de una manera indirecta la actividad muscular, ya que es capaz de determinar si el sistema nervioso está reclutando activamente las fibras de un músculo durante una tarea [3]. Durante la contracción se produce un potencial eléctrico que es medible en la membrana muscular, como se muestra en la Figura 2.1. Este fenómeno surge en unidades celulares llamadas motoneuronas [2], que a su vez se organizan en unidades motoras (las cuales llegan a establecer contacto con un promedio de 150 fibras musculares). Lo interesante de esto es entender que las señales electromiográficas están compuestas por cada uno de estos potenciales y, por ello, es posible descomponer las señales para obtener la actividad existente en cada una de estas organizaciones celulares.

Aplicado al control de prótesis, es importante reconocer y estudiar las zonas en donde se



Fuente: Boston University

Figura 2.1: Potencial de acción en la membrana muscular [3].

van a hacer las lecturas. Esto dependerá del tipo de amputación, del número de señales que queramos usar para el control e incluso de características como el porcentaje de grasa que tiene el paciente en cuerpo y la edad, entre otros.

Aunque las prótesis ideales son aquellas que permitirían realizar todos los movimientos de la mano, en la práctica los movimientos más utilizados son los de tipo “pinza” que consisten en abrir y cerrar la mano [11]. Más concretamente, como mostraremos adelante en esta memoria, nos hemos centrado en el desarrollo de una prótesis con 2 sensores, para lo cual mediremos la actividad del flexor carpi radialis y el extensor carpi radialis longus. Estos músculos ubicados en el antebrazo están implicados en varios movimientos de la mano. El flexor carpi tiene la acción de flexor principal de la muñeca, con tendencia a su abducción y pronación. El extensor carpi radialis longus, en la articulación del codo realiza una ligera flexión y en la de la muñeca una extensión, también colaborando en el cierre del puño y desviación radial. Con esta colocación de los sensores, buscaremos extraer características de la flexión y extensión de la muñeca y, si es posible, también del cierre del puño [12].

En esta zona del antebrazo podríamos diferenciar las acciones incluso provenientes de cada dedo, ello implicaría mayor y mejor sensorización y una capacidad de inteligencia superior en el controlador para distinguir las señales provenientes en cada movimiento. Esto sería lo más interesante si queremos imitar al máximo el comportamiento de una mano. Sin embargo, nos encontraríamos con ciertas restricciones, como el aumento del coste, un período mayor de adaptación y entrenamiento del paciente e incluso, un incremento en el tamaño o peso de la implantación.

## 2.2. Revisión del Hardware

La elección del hardware es clave para crear un sistema basado en EMG y debe tener en cuenta tanto la parte de garantizar la correcta adquisición de las señales como aquella relacionada con las características de los usuarios que van a utilizar la prótesis. A continuación, presentamos los aspectos más destacables del estudio que se ha realizado en este TFG para decidir el hardware a emplear en el desarrollo.

### 2.2.1. Electrodos

Los electrodos son como pequeños micrófonos que se utilizan para escuchar a los músculos. Existe una gran variedad de estos en el mercado, por lo que es necesario primero entender el uso que se le da a cada tipo para obtener la mejor calidad de señales EMG superficiales [12].

En cuanto a formas y tamaño, la elección dependerá de la zona muscular o músculo que queramos monitorizar y de la distancia entre electrodos que apliquemos. Lo más común son entre 0.5 cm y 1 cm de diámetro y distancia de 1 cm y 3 cm entre los mismos. Por ejemplo, si quisiéramos medir la actividad en un músculo específico en la cara de una persona, lo mejor sería utilizarlos del menor diámetro posible, una colocación precisa y con una separación de 1 cm. En cambio, si lo que buscamos es extraer la actividad de una zona como la espalda en la que hay varios músculos, sería necesario utilizar electrodos con la mayor superficie de contacto y posicionados a una distancia mayor entre ellos [12].

En cuanto a su composición, lo más usual es utilizar electrodos hechos de un plástico impregnado en plata que estará recubierto en una fina capa de cloruro de plata para ayudar a estabilizar los potenciales eléctricos en la piel. Se puede diferenciar entre 2 tipos: electrodos en seco y húmedos. La diferencia radicará en la utilización de un gel conductor entre el electrodo y la piel, que ayuda a conducir mejor la carga eléctrica. En general, los más cómodos de utilizar son los secos, ya que permiten una fácil interacción y se pueden recolocar sin mucho problema, lo cual nos será muy útil porque el posicionamiento de los mismos es crucial (y recalcamos esto último porque ha dado muchos dolores de cabeza).

Sobre el formato en el que suelen venir, opinamos que ya es hora de reinventarlos un poco. Para poder leer varios músculos en una misma zona su colocación puede llegar a ser bastante engorrosa.

Para las mediciones, comúnmente se hace uso de dos emisores (uno en cada extremo del músculo a medir) y masa (normalmente ubicada en zona de hueso) que permite extraer una señal más limpia. A modo de ejemplo, en la Figura 2.2 puede verse la colocación de cuatro electrodos húmedos en la zona del brazo correspondiente a los músculos flexores y extensores de muñeca (de los cuales hablamos en la anterior sección). La figura de arriba muestra los electrodos colocados sobre el flexor, mientras que en la figura de abajo están colocados sobre

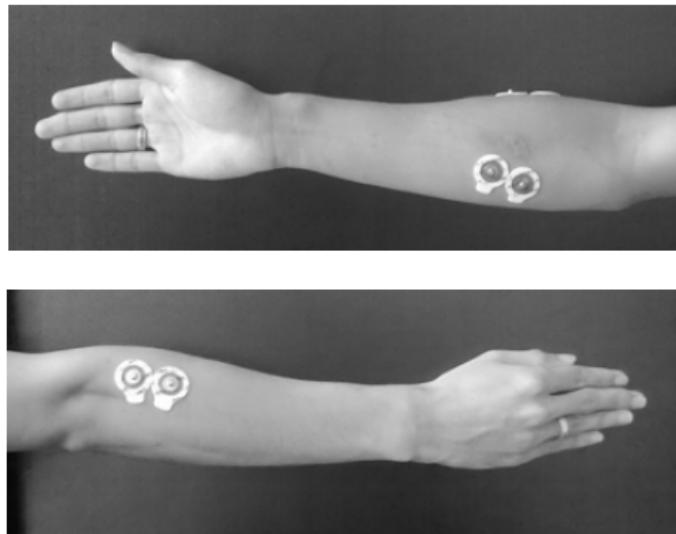


Figura 2.2: Colocación de los electrodos durante los experimentos [4].

el extensor.

Antes de comenzar el desarrollo, se analizaron distintos tipos de electrodos. La Tabla 2.1 recoge las características más importantes. Se pudo comprobar que ambos electrodos funcionan correctamente cuando se integran con las placas EMG. El electrodo pasivo de la marca Olimex tiene un tamaño y coste mayor, pero el hecho de que sea seco permite una colocación más rápida y limpia. Dado que solamente se colocarán en 2 zonas, el precio final es más que razonable para este proyecto. En base a este estudio, hemos decidido utilizar los de la marca Olimex.

Modelo	Fabricante	Tipo	Dimensión	Precio
Electrodos pasivos de ECG y EMG	Olimex	Seco	1cm	10€
Electrodos con parches desechables para ECG y EMG	Desconocido (China)	Húmedo	0.5cm	5€

Tabla 2.1: Electrodos utilizados en este proyecto.

### 2.2.2. Placas EMG

¿Cuál sería entonces la forma fácil de leer estas señales? A día de hoy el primer recurso de información al que se acude para responder algo así es Internet. Indagando, se puede encontrar que existen algunos fabricantes estadounidenses de circuitos impresos (PCB) para desarrollo

que ofrecen soluciones de difícil acceso y a alto coste, lo cual no se adapta al objetivo marcado en este TFG. Como ejemplo, se pueden consultar los detalles en las páginas web de Ossur [13] y de Biosignal R&D [14].

Existe un canal de Youtube llamado “Biomakers” [15] con tutoriales muy buenos sobre todo tipo de señales biológicas (musculares, cerebrales, etc.) donde hasta enseñan los diseños para la circuitería necesaria en la medición. Por ello, comenzaremos esta sección hablando a grandes rasgos de la composición de estos, ya que sorprende lo sencillos que son.

El principio básico de un sensor es mapear una magnitud real como la luz o la temperatura en funciones matemáticas. Extraemos valores del mundo que nos rodea para poder representarlos de forma transductible. En el caso del EMG, al estar leyendo los pocos milivoltios que se generan durante la actividad muscular, entre nuestra señal y sensor no existirá la necesidad de hacer transformaciones entre magnitudes, ya que simplemente servirá como amplificador de la actividad para obtener una señal manipulable.

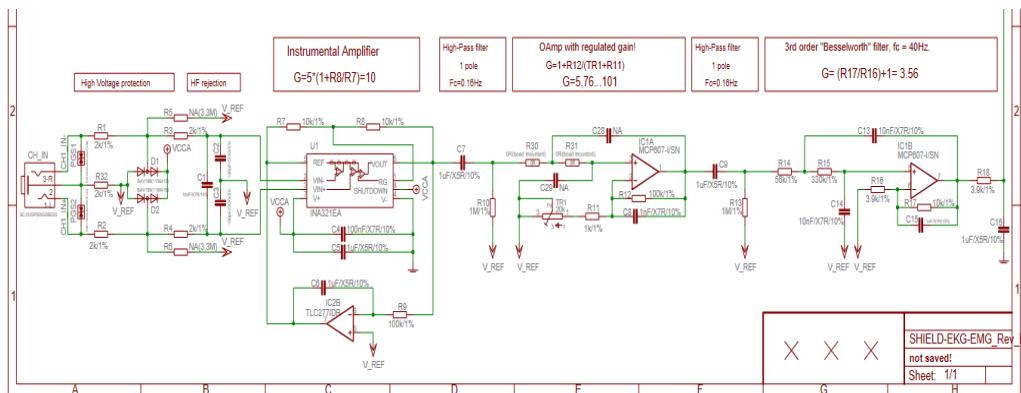


Figura 2.3: Esquemático de la placa Olimex [5].

El circuito de la Figura 2.3 hace referencia a la placa SHIELD-EKG-EMG de Olimex [5]. Como su nombre indica se trata de un “escudo” que se puede apilar sobre las placas de tipo Arduino Uno. Este cuenta con todos los componentes para ser calibrada y conectada a la misma y permite la lectura de varios canales. Esta parte de su esquemático se corresponde con la del sensor electromiográfico.

Empezando desde la izquierda del todo en el esquemático, vemos que existe una conexión “CH\_IN” en donde tenemos 3 terminales de conexión: L, R y Masa, correspondiéndose cada uno con los terminales en los electrodos. A grandes rasgos, lo que podemos ver es que las corrientes recogidas por estos elementos irán pasando por varias fases de amplificación y filtrado. Comenzando por componentes de protección, pasamos a una etapa donde rechazaremos frecuencias altas y, a continuación, amplificaremos la señal mediante un amplificador

operacional de cierta ganancia. Seguimos entonces con etapas de filtrado para la frecuencia de 0 Hz característica en la corriente continua para, finalmente, llegar a una etapa que merece cierta explicación.

En este caso se trata de una fase en donde utilizaremos un amplificador de ganancia regulada. Y ¿por qué regulada? Bien, durante la sección de conceptos biológicos se explicó que este tipo de sensores va a estar afectado por ciertas características físicas de la persona y, puede ocurrir, que no tenga la sensibilidad suficiente para recoger señales desde la superficie (ya sea por el índice de grasa que separa el sensor y el músculo o también por la poca masa muscular de la zona). Por esto mismo, la PCB de Olimex nos permite adaptarnos a cualquier situación. Sin embargo, la calibración para esta placa, en nuestra opinión, resulta un poco engorrosa, ya que se hace de forma física y mediante una señal por ancho de pulso (PWM) que genera, por lo que sería interesante que se pudiese hacer de manera completamente automática, mediante software.

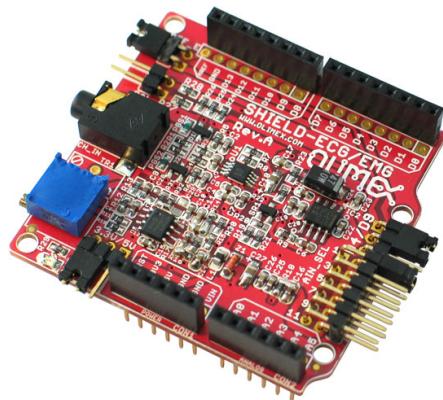


Figura 2.4: Sensor EMG de Olimex [5].

La Figura 2.4 muestra la placa. La señal en crudo que proporciona no permite una fácil interpretación de lo que está ocurriendo y presenta ruido en su información, es una onda completamente positiva y los valores durante una contracción o el reposo son casi indistinguibles. En esta situación sería prácticamente imposible extraer características para clasificar movimientos. La documentación existente no ayuda a entender la correcta utilización de la placa para leer EMG, ni explica el posible origen del ruido.

Sin embargo, encontramos por Internet una librería en C++ para filtrar señales EMG que fue probada en este TFG con el sensor de Olimex. El filtro descrito es de paso banda entre los 20 y los 150 Hz, dando como resultado una onda con componente positiva y negativa que me sugirió la idea aplicar un rectificado completo para extraer la potencia de la señal (concepto teórico que se explicará más adelante) y con la que se consiguieron buenos resultados finales.

La segunda placa analizada es de origen chino y su fabricante es desconocido, ya que se

compran desde un suministrador que no ofrece información. Según se puede leer por Internet, parece que podría tratarse de un caso de “copia” en la que basan sus diseños según modelos de otros fabricantes. Se puede encontrar en Aliexpress por un precio de 15 €, viene en un pack con los electrodos, parches y sensor. La [Figura 2.5](#) muestra una foto de la placa y, como puede verse, tiene un tamaño muy adecuado para realizar un desarrollo cómodo con facilidad de colocación y transporte.

Esta placa necesita una alimentación de +9 V y -9 V, utiliza la referencia positiva y negativa para hacer operaciones, la salida la da por el terminal de “SIG” y se trata de una señal analógica de entre 0 V y 3 V, es decir, completamente positiva (esto tiene sentido porque el Arduino es incapaz de leer voltajes negativos en sus terminales analógicos).

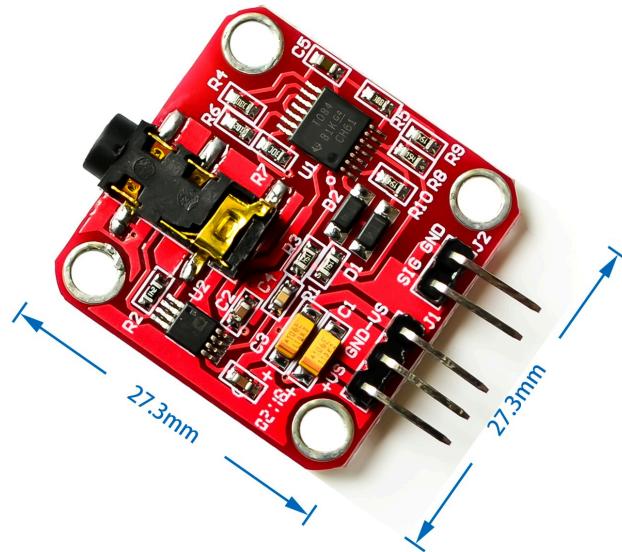


Figura 2.5: Sensor EMG de origen chino [6].

El montaje se puede ver en la [Figura 2.6](#) y es bastante trivial, lo puede hacer cualquiera en casa con un kit de principiantes para electrónica. Se puede ver que con 10 cables estaría todo montado, por lo que sería muy fácil con este tamaño e incluso un Arduino del tipo Nano, diseñar una carcasa en una impresora 3D para acoplarlo todo de forma cómoda.

Después de habernos colocado los electrodos, resulta sencillo comenzar a leer los primeros valores relacionados con la actividad muscular. Para ello, configuramos el puerto serial en el código del Arduino y, dentro del bucle “loop”, leemos las entradas analógicas en las que están conectadas los electrodos. La implementación de dicho programa es sencilla, por lo que no se detallará en esta memoria. Sin embargo, dicho código fuente se encuentra disponible en el

repositorio de GitHub [16].

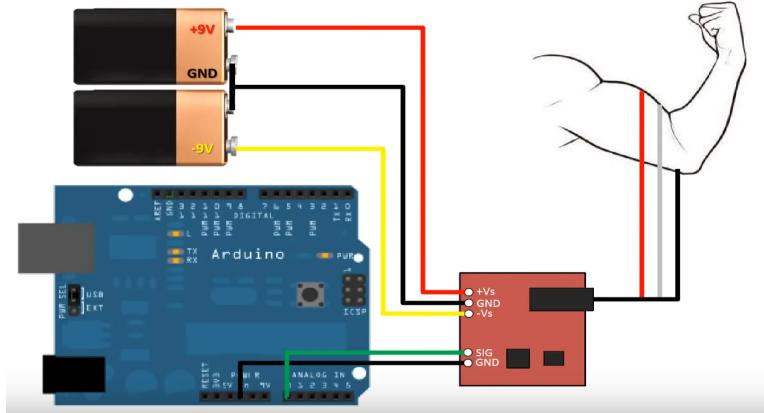


Figura 2.6: Esquema del montaje para la placa china [6].

La pruebas realizadas y los desarrollos que se presentan en este proyecto funcionan indistintamente con la placa Olimex y la de origen chino: los 2 sensores devuelven el mismo resultado y en esencia lo que las diferencia es su formato físico. La implementación del sistema en este trabajo es lo suficientemente abstracta como para que exista modularidad entre capa hardware y software. Lo único a destacar es que la placa Olimex solo funcionaría en un escenario en el que se pueda apilar sobre un Arduino del tipo Uno. Por otro lado, la placa adquirida en Aliexpress, presenta un precio que es muy accesible, el tamaño y utilidad ofrecida es muy buena, además, es fácil de poner a funcionar y también permite integración con cualquier otro socket con entrada analógica en su circuitería. Como podemos observar, cada PCB tiene sus ventajas y desventajas.

Existen también 2 placas estadounidenses que fueron consideradas para el desarrollo de este trabajo. El sensor Myoware [7] de Advancer Technologies y Gravity [8], de OYMotion.

En la Figura 2.7 tenemos la placa Myoware. Según vemos en la página web del fabricante, funciona con una sola alimentación de entre +3.3 V y +5.0 V. Tiene la capacidad de poder elegir entre una salida analógica que se correspondería con la señal muscular completamente rectificada o en crudo (conceptos que explicamos durante el capítulo 3). Se puede también, de forma física, ajustar la ganancia del equipo. Otra cualidad de la misma es el formato en el que se presenta, lleva en su carcasa soldados los electrodos. Su diseño nos permitiría usarla en cualquier escenario, acoplada a cualquier tipo de microcontrolador que sea capaz de alimentarla y de leer su señal analógica. Sin embargo, este sensor fue descartado por ahora debido a los sobre costes en su importación.



Figura 2.7: Sensor Myoware de 1 canal [7].

La otra PCB, el sensor Gravity, se puede ver en la Figura 2.8. Funciona con la misma alimentación que se utiliza para la anterior e, igualmente, se podría acoplar con cualquier socket que sea capaz de suministrarle energía y leer señales analógicas. Aunque, en este caso, la placa Gravity solo es capaz leer la señal muscular rectificada. Su diseño consta de 2 partes: la PCB con la circuitería sensorial y los electrodos en seco. Fue igualmente descartada por culpa de los sobrecostes que se generaban durante su importación.

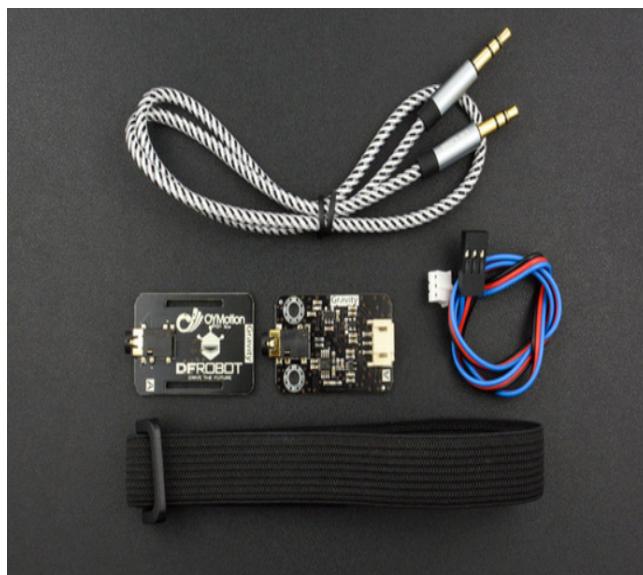


Figura 2.8: Sensor Gravity de 1 canal [8].

Finalmente, se intentó utilizar la PCB que se muestra en la [Figura 2.9](#). Esta placa tiene un sensor que podría también utilizarse para este propósito. Se trata de la placa Cyton de Open BCI [9] y actualmente se está usando en proyectos universitarios relacionados con electroencefalografía (EEG). Tiene soporte software, un programa y librerías para leer y analizar señales biológicas. Cuenta también con varios canales de entrada en los que se colocarían los electrodos. Sin embargo, estas conexiones no son compatibles con los electrodos utilizados durante este proyecto, a causa de esto, la placa fue descartada por ahora en los experimentos de este trabajo.

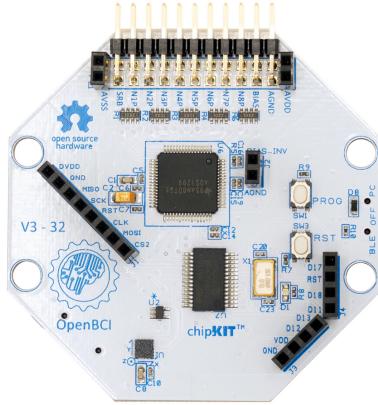


Figura 2.9: Sensor de 8 canales, Cyton de Open BCI [9].

### 2.3. Desarrollos conocidos

Durante la introducción se comentó que era necesario hacer un análisis del mercado actual y usos de esta tecnología. Es un tema que podría ser tratado durante capítulos, pero no es lo que pretendemos. Este documento tiene el propósito de servir como una guía sencilla de seguir, es decir, permitir al lector profundizar más en los conceptos tratados en esta memoria. Será entonces más interesante hacer un resumen del panorama actual de esta tecnología en donde destacaremos desarrollos que han tenido gran influencia sobre este proyecto.

La empresa española MDURANCE [10] suministra este tipo de equipamiento a clínicas tanto deportivas como de terapia ocupacional. Ha conseguido una buena recepción entre sus clientes, ofreciendo tanto hardware como software listo para usar por los especialistas en sus sesiones.

Es innegable que este tipo de equipamiento no es algo nuevo y que ya se utilizaba desde hace tiempo en el área del deporte e investigación. Igualmente, hay que decir que se trata de material con un coste que pocos sitios y particulares pueden afrontar, por lo que es común

replantearse su verdadera utilidad. Otro motivo, no menos importante, es que de nada sirve tener equipamiento para lecturas EMG si luego no tenemos un buen soporte software que facilite la interacción con estas señales. Al final, lo interesante es la información que se puede extraer después de los distintos procesados (materia que ya no está al alcance de cualquiera). Esto mismo, es lo que proporciona MDURANCE, un producto de principio a fin, comprar y usar.

Como se puede ver en la [Figura 2.10](#) y [Figura 2.11](#), el equipamiento se utiliza para obtener lecturas de la actividad muscular durante ciertos movimientos y así ofrecer al especialista información más detallada e individual sobre sinergias musculares, descompensaciones y fatiga, entre otros.

Lo más interesante sobre el producto ofrecido aquí es la escalabilidad para implementar un sistema que no solo se base en la electromiografía y crezca hacia un asistente clínico con la capacidad de sacar conclusiones en base a diferentes datos recolectados durante las sesiones. Ahora mismo, cuentan también con *Videofeedback* y monitorización del progreso.

El problema actual que vemos en su modelo de negocio es en la adquisición de su producto. Es necesario primero contactar con ellos y esto da a entender que trabajan bajo demanda y los precios seguro pueden asustar a más de uno. Después de tener alguna conversación con ellos, posiblemente esta situación se desarrolle principalmente por culpa de hardware suministrado desde terceros, si a esto le sumamos las pocas empresas en este nicho de equipamiento y por lo tanto la baja competencia, los costes seguramente sean altos.

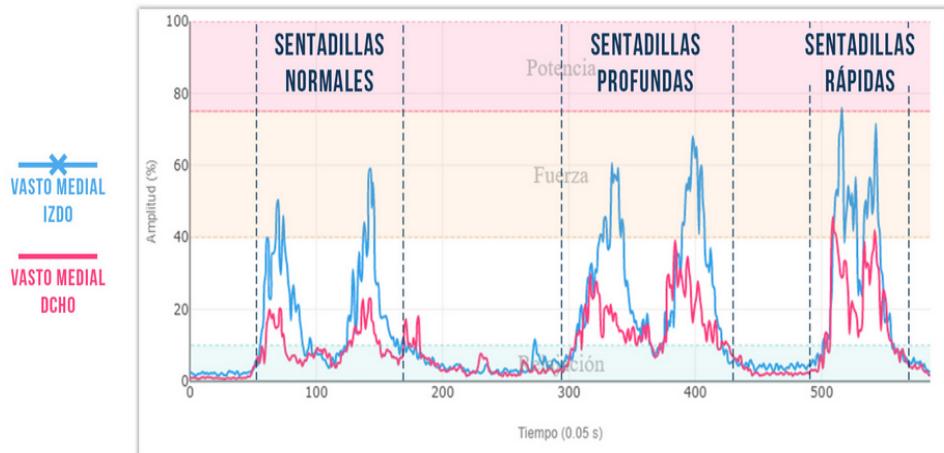


Figura 2.10: EMG durante la sentadilla [10].

Finalmente, comentar que el electromiógrafo de este fabricante es de 4 canales y con tecnología bluetooth, tiene una autonomía de 12h y una frecuencia de muestreo a 1024Hz. Estas

características en principio se podrían conseguir con nuestra placa de origen chino. En la anterior sección se habla sobre las virtudes de la PCB comprada en Aliexpress por su tamaño, precio y la posibilidad que tiene para configurarse dentro de una caja para así, incluso conseguir construir un electromiógrafo con las mismas dimensiones que el de MDURANCE. Con respecto a la conectividad, actualmente es muy fácil encontrar microcontroladores con conectividad inalámbrica y también existe la posibilidad de configurar un Arduino Nano para enviar datos mediante bluetooth a una frecuencia de muestreo parecida al de su producto.



Figura 2.11: Activación de trapecio en paciente sano versus lesionado [10].

Otro nicho interesante que se puede ver en el mercado es el uso de EMG como interfaz hombre-máquina, donde se utilizaría como método para controlar elementos externos o cualquier tipo de máquina mediante la caracterización de movimientos musculares del usuario. Este principio es el mismo utilizado para el control de prótesis. Un ejemplo sencillo de esto se encontraría dentro de lo que conocemos como Realidad Virtual, en donde se busca sumergir a la persona en un mundo completamente digital y existiría un avatar controlado por los movimientos musculares de la misma. Los videojuegos también son un método interesante para introducir a la gente en este tipo de tecnologías y hacer más amigable la interacción durante los experimentos o rehabilitaciones. Se enseña de forma gráfica y amena de lo que se trata todo esto. Actualmente, por Internet, se puede encontrar bibliografía que trata el tema y algunos desarrollos, demos, etc. Pero ninguno de estos proyectos es destacable ni famoso, se trataría de un escenario que está actualmente poco explotado, pero el cual podría funcionar muy bien y es utilizado en este trabajo [17].

En cuanto al mercado de prótesis de mano, se hace en falta más divulgación informal que acerque esta tecnología a cualquier tipo de público. Casi toda la información hace referencia a artículos académicos o científicos, los cuales no son de acceso abierto en la mayoría de ocasiones. Por el contrario, en el canal de Youtube "Biomakers" que habíamos mencionado en la anterior sección, divultan y enseñan esta tecnología de forma gratuita. Se puede incluso encontrar material y cursos sobre diferentes ramas de la bioingeniería. El único inconveniente de esto, podría ser que no suelen entrar en profundidad sobre los desarrollos y dejan para sus cursos (que no son precisamente baratos) cualquier información sobre código o diseños. Destacar que también tienen una tienda web en donde venden material para la construcción y puesta en funcionamiento de brazos robóticos, sensores, material para impresión 3D, etc. [18].

Alt Bionics [19], empresa estadounidense creada con el objetivo de construir prótesis baratas, actualmente tienen prototipos bastante avanzados y buscan abrirse al mercado con un producto que saldría en los 3500\$. Decir que son muy activos por redes sociales y apoyan a la comunidad de investigadores pretendiendo en un futuro ofrecer divulgación accesible sobre el tema.

Es imposible no ver el auge de la robótica en los últimos años y esto beneficia mucho a la construcción de este tipo de implantaciones. Bastantes empresas en este nicho trabajan en la fabricación de exoesqueletos, brazos robóticos y ofrecen el producto acabado o kits con los diseños 3D y tutoriales para el ensamblado. Esto evitaría bastantes molestias, ya que se trata de una parte importante en la manufactura para este tipo de sistemas en los que igualmente, quedaría trabajar en el software del mismo. Un ejemplo de empresa siguiendo este modelo de negocio sería YouBionic [20]. Otra opción actualmente, sería descargar por internet diseños Open Source o a muy bajo precio para implantaciones de cualquier tipo. Con la posibilidad de ajustarlos individualmente a un paciente en concreto, esta es una de las mejores opciones para construir un primer prototipo.

Respecto a la situación de estos productos en España, actualmente existe una asociación a nivel nacional [21] para gente con este tipo de discapacidades. El gobierno tiene un programa para ofrecer de forma gratuita prótesis a pacientes amputados. En concreto están utilizando la prótesis Michelangelo [22], la cual devuelve una cantidad importante de movimientos. Sin embargo, se trata de una implantación que por su precio (alrededor de 60.000€) es difícil de adquirir por muchas personas.

Por último, nos gustaría destacar que los terapeutas nos han indicado que no todas las personas pueden acceder a este programa de prótesis gratuitas por la existencia de posibles incompatibilidades debido a las condiciones de la zona amputada, adaptación al miembro y por posibles problemas debido a la sensación de miembros fantasma. En nuestra opinión, la existencia de prótesis de bajo coste permitiría que más personas pudiesen probar si se sienten

comfortables con ellas.

## **2.4. Conclusiones**

En este capítulo se ha realizado una revisión de distintos productos tanto hardware como software relacionados con el desarrollo del sistema que presentaremos a continuación.

Al principio de este proyecto, solo se contaba con el sensor de Aliexpress y fue luego cuando tuvimos acceso a la segunda placa de Olimex. Por tanto, se realizaron 2 configuraciones buscando siempre adaptarnos a las necesidades del momento. Por otro lado, el resumen presentado en este capítulo de los desarrollos realizados en otros autores, nos ha permitido establecer ciertas características de las aplicaciones que permitan emplear un desarrollo basado en EMG a ámbitos diferentes a la construcción de prótesis. Así, a parte del modelo 3D de prótesis, en este TFG se ha integrado la parte hardware con algunos juegos conocidos.

## Capítulo 3

# Arquitectura Desarrollada

---

EN el capítulo anterior hemos hecho un planteamiento general y divulgativo sobre el estado actual de la electromiografía. Partimos, por tanto, de una base con conocimientos básicos sobre los principios biológicos, material para adquisición de señales e información sobre desarrollos u objetivos que se quisieran llegar a consumar en un futuro.

Comenzaremos este capítulo hablando de las tecnologías empleadas en el desarrollo de todo el sistema. Todo esto incluye la parte de programación de Arduino y la de desarrollo de la interfaz gráfica para integrar todos los módulos de la arquitectura.

La Figura 3.1 muestra el esquema de la arquitectura cuyos módulos serán descritos en las secciones que presentaremos a continuación: adquisición de señales, procesado de señales y aplicación. A modo de ejemplo, se mostrarán los resultados de algunas pruebas que se hicieron para verificar el correcto planteamiento del sistema.

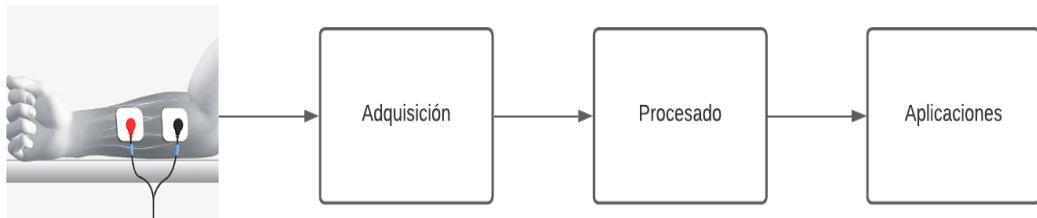


Figura 3.1: Esquema de la arquitectura.

### 3.1. Tecnologías

Durante esta sección comentaremos y describiremos las tecnologías utilizadas a lo largo de este trabajo.

### 3.1.1. Arduino

Arduino [23] es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Utiliza el lenguaje de programación Arduino (basado en Wiring) y el Software Arduino (IDE), basado en Processing.

A lo largo de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores y profesionales) se ha reunido en torno a esta plataforma de código abierto. Sus contribuciones se han sumado a una increíble cantidad de conocimiento accesible que puede ser de gran ayuda tanto para principiantes como para expertos.

Resulta imprescindible destacar la gran importancia que tiene para este proyecto la existencia de la comunidad de desarrolladores que conforman Arduino. Sin la posibilidad del acceso a toda la documentación disponible, para el uso y comprensión de los lectores analógicos disponibles en la placa y comunicaciones serial, habría sido imposible comenzar fácilmente con el desarrollo.

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta desde placas simples de 8 bits hasta productos para aplicaciones de Internet of Things, wearables, impresión 3D y entornos integrados. Todas las placas Arduino son completamente de código abierto, lo que permite a los usuarios construirlas de forma independiente y eventualmente adaptarlas a sus necesidades particulares. El software también es de código abierto y crece gracias a las contribuciones de los usuarios de todo el mundo.

### 3.1.2. Python

Para el desarrollo del proyecto se eligió Python. Fue creado a principios de la década de 1990 por Guido van Rossum en Stichting Mathematisch Centrum en los Países Bajos, como sucesor de un idioma llamado ABC. Guido sigue siendo el autor principal de Python, aunque incluye muchas contribuciones de otros. Todas las versiones de Python son de código abierto. Históricamente, la mayoría de las versiones, también han sido compatibles con la Licencia Pública General (GPL) [24].

Es un potente lenguaje de programación interpretado. Tiene estructuras de datos eficientes de alto nivel y un enfoque simple pero efectivo para la programación orientada a objetos. La elegante sintaxis y la escritura dinámica de Python, lo convierten en un lenguaje ideal para la creación de scripts y el desarrollo rápido de aplicaciones en muchas áreas en la mayoría de

las plataformas [24].

Durante el trabajo se utilizaron librerías de código Open Source para el desarrollo de la aplicación. Las más destacables fueron las siguientes:

- Matplotlib.
- Numpy.
- Serial.
- Scikit-Learn.

Matplotlib [25] es una librería que permite crear visualizaciones estáticas, animadas e interactivas. Fue creada por John Hunter (1968-2012), quien, junto a muchos contribuyentes, han dedicado tiempo y esfuerzo a producir tal pieza de software utilizado por miles de científicos en todo el mundo. En nuestro proyecto, fue utilizado principalmente para representar gráficamente las señales EMG y sus características.

NumPy [26] es un proyecto de código abierto que tiene como objetivo habilitar la computación numérica con Python. Es por esto que nos fue útil para crear y manejar las estructuras de datos que conforman las señales musculares y así poder extraer parámetros (como la media mediante) funciones auxiliares que contiene la librería.

Fue creado en 2005, basándose en el trabajo inicial de las bibliotecas Numeric y Numarray. NumPy siempre será un software 100 % open source, de uso gratuito para todos y publicado bajo los términos liberales de la licencia BSD modificada. Se desarrolla de forma pública en GitHub, a través del consenso de desarrolladores y de la comunidad científica en general. Consta también de un enfoque de gobernanza constituido por varios miembros en forma de consejo.

Pyserial [27] se trata de una librería que encapsula el acceso a las conexiones de tipo serie entre dos sistemas. Provee de soporte para cualquier versión de Python corriendo tanto en Windows, OSX, Linux o IronPython. En nuestro caso, utilizamos sus funciones para establecer las comunicaciones con el Arduino, sincronizarlo con nuestro programa en Python y, finalmente, leer las muestras recogidas por los sensores. Este proceso fue muy fácil de implementar gracias a toda la documentación disponible que hay sobre este proyecto. Se trata de software completamente gratuito y desarrollado públicamente en GitHub.

Scikit-learn [28] es una librería OpenSource utilizada para Aprendizaje Máquina en donde podemos encontrar algoritmos para entrenar y evaluar modelos o preprocesar conjuntos de datos para su posterior utilización. Este proyecto de David Cournapeau nació en 2007 durante la “Google Summer of Code”. Años después, Matthieu Brucher seguiría con él como parte de su proyecto para la tesis. En 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort y

Vincent Michel de INRIA tomaron el liderazgo del proyecto y sacaron a la luz las primeras publicaciones en febrero de 2010.

Durante este trabajo conseguimos, gracias a su interfaz de programación de aplicaciones (API) para máquinas de vectores de soporte, clasificar movimientos de la mano y utilizarlos como una interfaz-hombre máquina. La librería se encargó de entrenar y validar el modelo, nosotros simplemente ajustamos algunos de los parámetros necesarios para obtener los mejores resultados.

### **3.1.3. Blender**

Además, para el modelado 3D decidimos utilizar la herramienta Blender [29]. Se trata de una suite de creación 3D gratuita y de código abierto. Es compatible con todo el proceso de construcción en 3D: modelado, montaje, animación, simulación, renderizado, composición y seguimiento de movimiento, incluso edición de vídeo y creación de juegos. Los usuarios avanzados emplean la API de Blender para scripts de Python para personalizar la aplicación y escribir herramientas especializadas (cosa que utilizamos actualmente en este TFG); a menudo, estos se incluyen en las versiones futuras de Blender. Es ideal para individuos y pequeños estudios que se benefician de su proceso unificado y de desarrollo receptivo. Esta herramienta cuenta también con un escaparate donde hay disponibles ejemplos de muchos proyectos basados en Blender.

Es multiplataforma y se ejecuta igualmente bien en ordenadores Linux, Windows y Macintosh. Su interfaz usa OpenGL para brindar una experiencia consistente. Para confirmar la compatibilidad específica con su sistema, se puede comprobar la lista de equipos probados por los desarrolladores.

Como proyecto impulsado por la comunidad bajo la licencia GPL, el público está facultado para realizar cambios pequeños y grandes en la base del código, lo que conduce a nuevas funciones, correcciones de errores sensibles y una mejor usabilidad. Blender no tiene precio, pero se puede invertir en él, participar y ayudar en su desarrollo y proyecto.

## **3.2. Adquisición de señales**

La primera etapa del sistema mostrado en la [Figura 3.1](#) es la correspondiente a la adquisición de las señales. Para leer señales EMG, contamos con 2 configuraciones (ver la [Figura 3.2](#)): la primera en la que utilizamos las placas Olimex y la segunda con las placas de Aliexpress. En ambos casos, se emplearon los electrodos pasivos de Olimex. Como puede verse, el montaje correspondiente a la placa Olimex (foto de la izquierda) resulta mucho más compacto y atractivo. Sin embargo, no tuvimos acceso a estos sensores hasta finales del desarrollo y, por lo tanto, los primeros experimentos fueron realizados por el alumno con la configuración traída

de Aliexpress.

Como comentamos en el capítulo anterior, los sensores de Aliexpress producen una señal analógica positiva que será leída por las entradas de conversión analógica digital del Arduino Uno. Esta placa cuenta con un conversor analógico-digital (ADC) del tipo conversor por sucesiones aproximadas (SAR) [23]. Este conversor funciona mediante un simple algoritmo de búsqueda binaria que usa un comparador para estrechar sucesivamente un rango que se aproxime al voltaje de entrada. En cada paso, se compara el voltaje de entrada con una referencia interna producida por el mismo Arduino. Este valor puede representar el punto medio del rango de voltaje seleccionado en cada paso del proceso, la aproximación finalmente se guarda en un registro. Este ADC tiene una resolución de 10 bits y los resultados estarán acotados entre 0 y 1023.

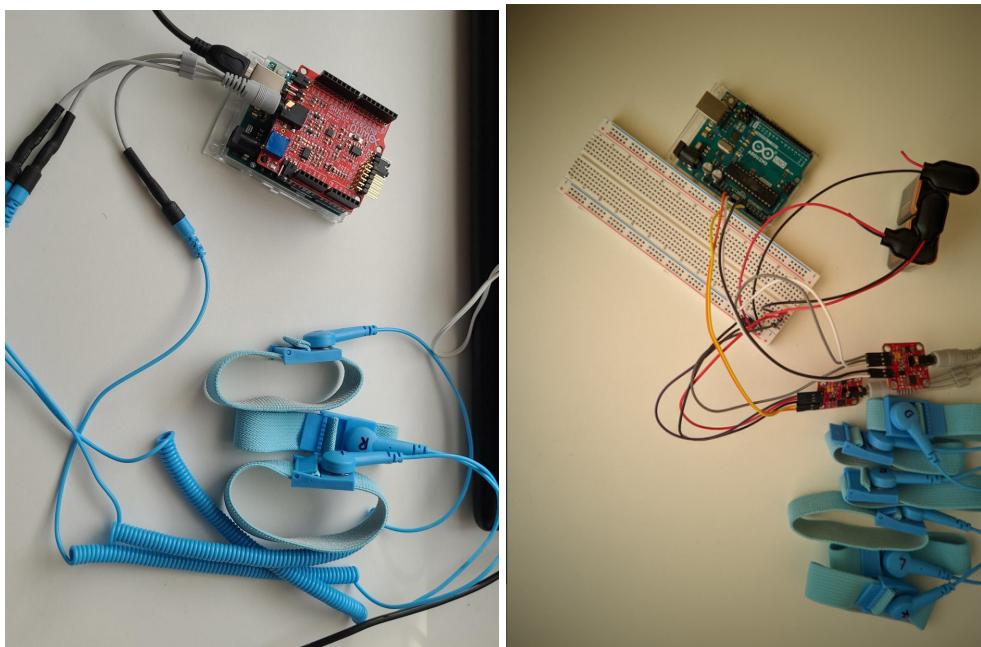


Figura 3.2: Montajes de las placas: Olimex (izquierda), placa de Aliexpress (derecha).

En el capítulo anterior ya hemos hablado de los montajes de la Figura 3.2, aquí nos gustaría hacer énfasis de los parámetros que se configuran para las mediciones. Para todos los experimentos configuramos una frecuencia de muestreo a 1000 Hz y BaudRate de 115200 baudios para evitar un cuello de botella en la velocidad de transmisión para los datos. La razón del primer valor se basa en las siguientes dos ideas:

- Las frecuencias significativas en el EMG se encuentran entre los 0 Hz y los 500 Hz. Por lo tanto, para poder trabajar con dicho rango de frecuencias debemos establecer, como mínimo, una frecuencia de muestreo de 1000 Hz, evitando así el efecto de *aliasing* [12].

- Si se quieren obtener buenos resultados en la prótesis o en cualquier otro desarrollo, lo recomendable es que el tiempo de respuesta de la aplicación sea menor a 300 ms. Por lo tanto, en nuestro sistema hemos decidido analizar secuencias de 250 muestras, lo que se correspondería con un tiempo de respuesta aproximado de 250 ms [4].

Es importante tener en cuenta que, aunque buscamos buenos resultados con 250 muestras, es recomendable durante los experimentos capturar muchas más para tener de sobra en los análisis.

Para las mediciones finales, se creó una herramienta software en Python que se encarga de conectarse al Arduino, sincronizarse con él, leer 1000 muestras de hasta 2 canales y exportarlo todo a Comma Separated Values (CSV), formato estándar que será útil para alimentar las librerías de análisis o los modelos de clasificación que se crearon. Además, ofrece una interfaz gráfica simple en la que se puede gestionar el tipo de medición que estamos haciendo, llevar la cuenta de las mismas, enseñar gráficamente el resultado e incluso interaccionar con el usuario mediante un sonido para indicarle cuando debe realizar el movimiento.

### 3.2.1. Pruebas preliminares

Para ilustrar el funcionamiento de esta etapa de la arquitectura, mostraremos los resultados de varias pruebas realizadas por el estudiante con la configuración de sensores de Aliexpress. Recordamos entonces los 4 estados de la mano que pretendemos identificar:

- Flexión de muñeca.
- Extensión de muñeca.
- Reposo.
- Cierre de puño.

La Figura 3.3 muestra las señales obtenidas en los dos canales para tres movimientos: extensión de la muñeca, flexión de la muñeca, reposo y cierre del puño. Se trata de una figura en tiempo real. La señal azul se corresponde al primer canal situado en el músculo flexor carpi radialis y la roja al segundo canal, situado en el músculo extensor carpi radialis longus. Es importante observar que la escala se adapta a la señal, por lo que la subfigura de reposo tiene una escala mucho menor. Recordamos, que el ADC del Arduino tiene una resolución de 10 bits y por lo tanto como máximo nuestras señales alcanzarán las 1023 unidades.

Vemos fácilmente como para la extensión de la muñeca (a) la amplitud de la onda roja es mucho mayor a la azul, y viceversa, durante la flexión (b). La colocación de los electrodos en músculos que son antagonistas, nos permite hacer una clasificación espacial de los movimientos. Para el reposo (c), tenemos valores pico en las 2 señales que no superan las 240 unidades,

incluso en el flexor vemos valores muy próximos a 0. Por el contrario, durante el cierre del puño (d), los 2 sensores alcanzan valores máximos cercanos a 1023.

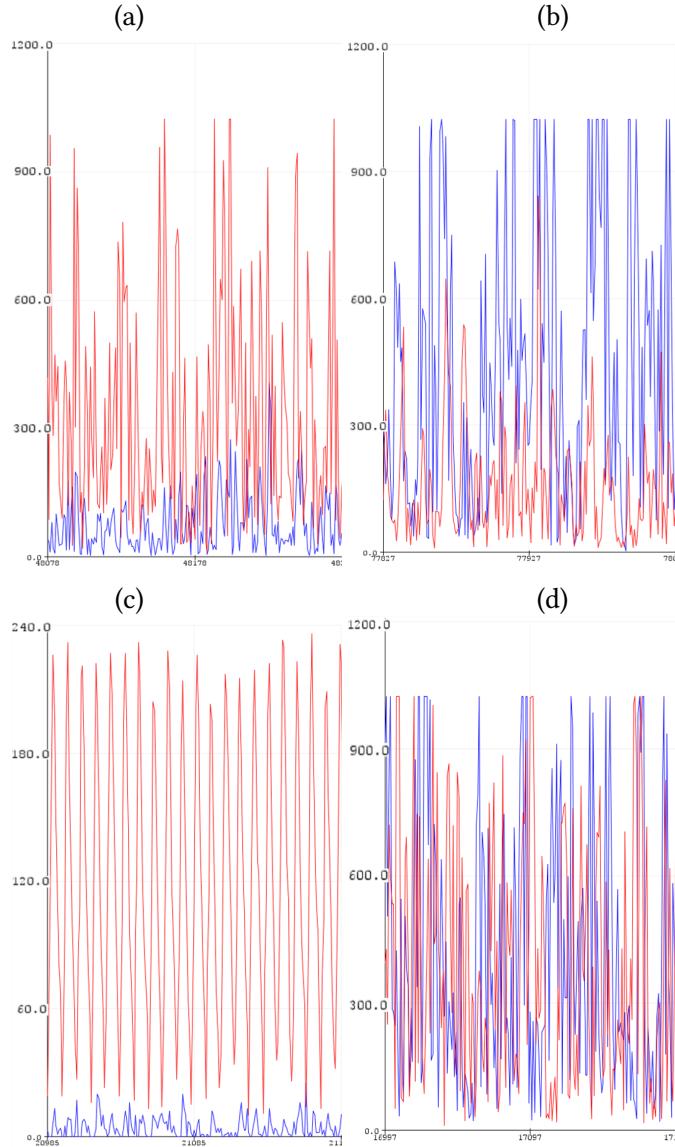


Figura 3.3: Salida del Serial Plotter en el IDE Arduino para (a) extensión, (b) flexión, (c) reposo y (d) cierre de puño.

Viendo esta primera ventana de muestras es fácil comprobar que nuestro planteamiento es correcto y que se pueden distinguir fácilmente los 4 estados.

### 3.3. Procesado de señales

Una vez adquiridas las señales, el siguiente módulo de la Figura 3.1 es el de procesado. Dicho módulo se encarga de extraer ciertas características de la señal obtenida que nos permiten determinar el movimiento realizado con los músculos. Por lo tanto, este procesado estará dividido en dos tareas principales: extracción de características y clasificación del movimiento realizado. Para ello, se han implementado las técnicas que explicamos a continuación.

#### 3.3.1. Extracción de características

La señal EMG obtenida y que veíamos en la prueba anterior, es un ejemplo de lo que se conoce como Raw EMG (EMG en “crudo”). Las señales en crudo contienen información esencial, pero en un formato que no resulta útil para detectar el tipo de movimiento realizado, ya que, como comentábamos durante la introducción, varía entre personas e, incluso, varía en función de la posición los electrodos. Es muy común que el siguiente paso entonces consista en rectificar la señal, es decir, en obtener el valor absoluto de la onda.

La rectificación puede ser de dos tipos, de media onda, en la cual se descartan los valores negativos y sólo se tienen en cuenta los positivos, o de onda completa, en la cual se obtiene el valor absoluto de los valores negativos. En nuestro caso la rectificación de onda completa es preferible para el posterior análisis que aplicamos. En la literatura científica también se ha empleado el rectificado de la EMG para detectar más fácilmente las frecuencias bajas (indicativo de la frecuencia de disparo de las unidades motoras), ya que estas no son fácilmente detectables [30]. Si bien es cierta esta afirmación, hay casos en los que esta técnica puede provocar que se detecten picos a bajas frecuencias que no están relacionados con el ratio de descarga. En definitiva, esto nos será de utilidad para, por ejemplo, analizar la amplitud en el dominio del tiempo.

En las señales EMG capturadas por los electrodos encontramos componentes aleatorios provocados por el ruido de fondo en la actividad muscular, causado por el mismo organismo, la colocación de los electrodos o incluso por el propio equipo de mediciones. Por lo tanto, su amplitud no puede ser definida con características específicas de la forma de onda (como el valor de pico), sino que necesita estimaciones estadísticas. Para calcular la amplitud del EMG, se pueden emplear dos métodos: raíz media cuadrática (RMS) o valor medio rectificado (ARV). El RMS es el método más empleado por su claro significado físico, mientras que el ARV simplemente provee la medida del área bajo la señal EMG [31].

La RMS es una medida estadística que busca recoger información en una serie de valores discretos o función matemática de variable discreta sin que afecte el signo de sus componentes, es decir, de cierta manera estamos aplicando a la vez el concepto de rectificado de onda al calcular esta característica. A partir de  $N$  muestras de una señal  $X$ , el valor del estimador

RMS se calcula haciendo la siguiente operación:

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N x(n)^2}, \quad (3.1)$$

donde  $x(n)$  es la n-ésima muestra de la señal  $X$ .

Por lo tanto, consideramos que el RMS es el método más adecuado para extraer las características de nuestro sistema. De esta forma, este valor RMS será utilizado para la determinación de los movimientos que realiza el usuario. Es decir, de cada secuencia de 250 muestras se extraerá su valor RMS para clasificarlo en un tipo de movimiento muscular.

### 3.3.2. Clasificación basada en umbral

Entendido ahora el concepto de RMS, en esta sección se va a explicar cuál va a ser su función y dónde vamos a aplicarlo. La idea consiste en reconocer el momento en el que se produce la contracción muscular, la cual representa el principio básico para la existencia del movimiento, y diferenciarlo del reposo. Utilizando solo 1 sensor, estos 2 estados de contracción o reposo dependen directamente de la zona en la que coloquemos los electrodos. En nuestro caso, al tenerlos en el antebrazo se producirá una contracción al realizar cualquier movimiento de la mano, como el cierre del puño, flexión o extensión de la muñeca, o incluso durante ciertos movimientos en los dedos.

Aunque el método basado en umbral puede ser utilizado para diferenciar dos estados cualquiera, en este proyecto lo hemos aplicado para clasificar los siguientes:

- Flexión de muñeca.
- Reposo.

Para reconocer estos 2 movimientos utilizaremos el RMS, ya que durante la contracción muscular se disparan las unidades motoras y la amplitud de la señal electromiográfica aumenta considerablemente. Entonces podremos fijar un valor umbral con el que podremos diferenciar si se está realizando un movimiento o se está en reposo. Este valor de umbral se determina durante una primera etapa de entrenamiento, donde el usuario realiza mediciones de prueba para calibrar el sistema.

El cálculo del umbral es muy sencillo. En nuestro caso, utilizaremos varias mediciones independientes de la señal EMG durante el reposo y durante la contracción. A partir de las muestras obtenidas en cada medición, calcularemos la RMS para después calcular la media aritmética por cada estado. Finalmente, el punto medio entre estos 2 valores funcionaría como umbral.

En base a nuestros experimentos, podemos decir que 10 mediciones de cada estado son suficientes para obtener buenos resultados. Aumentarlas en número no consiguió ofrecer mejora y, además, lastraba los experimentos alargando las sesiones.

Con este clasificador ya sería posible controlar una prótesis de tipo gancho, en donde la contracción (flexión de muñeca en nuestro caso) o aumento del RMS significaría el cierre de la mano en la implantación. En este caso, se ha explicado el cálculo del umbral de forma automática, pero es importante que el sistema también permita la calibración manual del mismo, ya que podremos encontrarnos con pacientes que tengan dificultades para alcanzar dicho valor, ya sea por fatiga del músculo, dolencias en la extremidad, etc.

### 3.3.3. Clasificación utilizando máquina de vectores de soporte

La técnica del umbral es fácil de programar y funciona bien en el caso de solo utilizar 1 sensor. Está claro que el funcionamiento es muy trivial, no tiene ningún tipo de complejidad, pero, ¿qué pasaría en el caso de usar más sensores? ¿Cómo gestionamos múltiples umbrales? Bien, en este proyecto llegamos a trabajar con 2 sensores como comentábamos en capítulos anteriores. Por lo tanto, ahora en vez de solo flexión y reposo tenemos 4 posibles estados, que a partir de ahora llamaremos “clases” de movimiento:

- Flexión de muñeca.
- Extensión de muñeca.
- Cierre de puño.
- Reposo.

De forma simplificada, podemos decir que la idea radica en una clasificación espacial en donde por la colocación de los electrodos y la naturaleza de estos movimientos, durante la flexión de la muñeca se “activaría” uno de los sensores mientras que el otro se mantendría en estado de reposo y, así mismo, durante la extensión ocurriría lo contrario. En el reposo, los 2 sensores estarían inactivos y, durante el cierre del puño, se activarían los 2.

Para modelar este comportamiento utilizamos varias mediciones de cada clase (en nuestras pruebas, fueron 15) y el algoritmo de clasificación basado en máquinas de vectores de soporte (SVM) y la característica RMS calculada sobre cada sensor. Este algoritmo dibuja un espacio vectorial con todas las mediciones de cada clase y traza planos (o hiperplanos) para delimitar zonas entre los movimientos, usa los “Vectores de Soporte” (referencia aquellos puntos que están más cerca de la frontera con la otra clase) para trazar estas zonas en las que agrupa las características del mismo tipo. Para ello utiliza diferentes kernels o núcleos que darán formas a las zonas según la naturaleza de nuestros datos. Aquí estudiaremos el kernel Lineal, Polinomial, Radial y de tipo Sigmoide[32].

También se apoya en un proceso de entrenamiento, en donde comprueba en base a un conjunto de puntos de test si las zonas dibujadas aciertan a la hora de clasificar estos datos de prueba y así, calcular cuánto de eficaz es el modelo.

El código desarrollado en este TFG es bastante simple. Se utilizó la librería de Scikit-Learn para entrenar y validar los modelos. Añadir que este tipo de clasificadores realiza una clasificación binaria por defecto, por lo que es necesario añadir pequeños trucos para tener varias clases de uso. Construye  $n_{clases} * (n_{clases} - 1)/2$  clasificadores, los entrena y aplica una función de decisión del tipo 1 contra 1 (algoritmo que puede ser transformado luego a 1 contra el resto en donde el número de clasificadores baja hasta llegar al número de clases).

### 3.3.4. Pruebas preliminares

Durante esta sección se adjuntarán las pruebas realizadas tanto con la técnica del umbral como con la máquina SVM. Recordamos que estas primeras pruebas fueron realizadas por el estudiante con la configuración de Aliexpress.

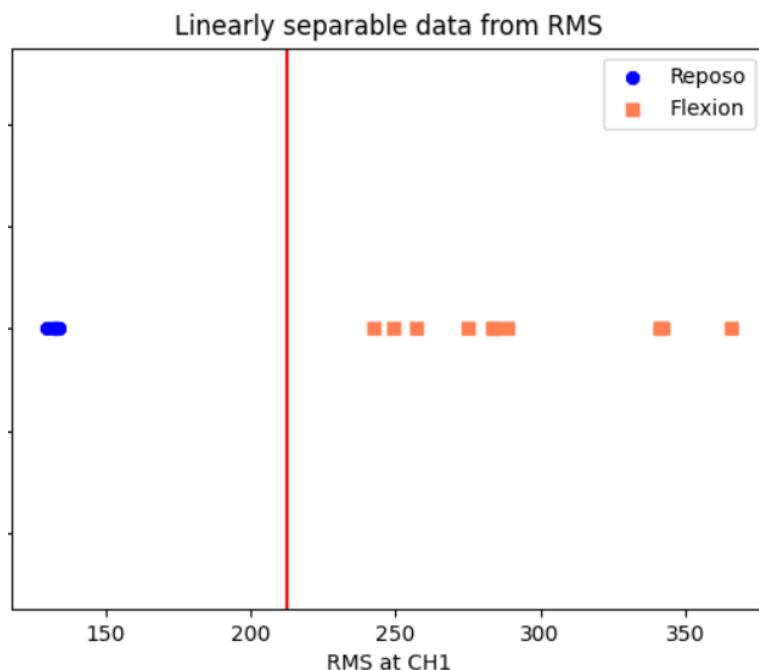


Figura 3.4: RMS calculado sobre las mediciones hechas para la técnica de umbral: reposo (azul), flexión (naranja) y linea roja el umbral calculado.

En la Figura 3.4 hemos querido representar los resultados del entrenamiento para la técnica del umbral descrita en la sección 3.3.2. El valor RMS de cada medición se puede ver en el eje X de la figura. La nube azul se corresponde a las 10 mediciones en reposo y las 10 naranjas son las obtenidas durante la flexión. Finalmente, la linea roja situada en un valor de 210 se

trata del umbral calculado por el sistema que nos permitirá hacer la clasificación entre los 2 movimientos. Dicho valor fue posteriormente utilizado durante el juego.

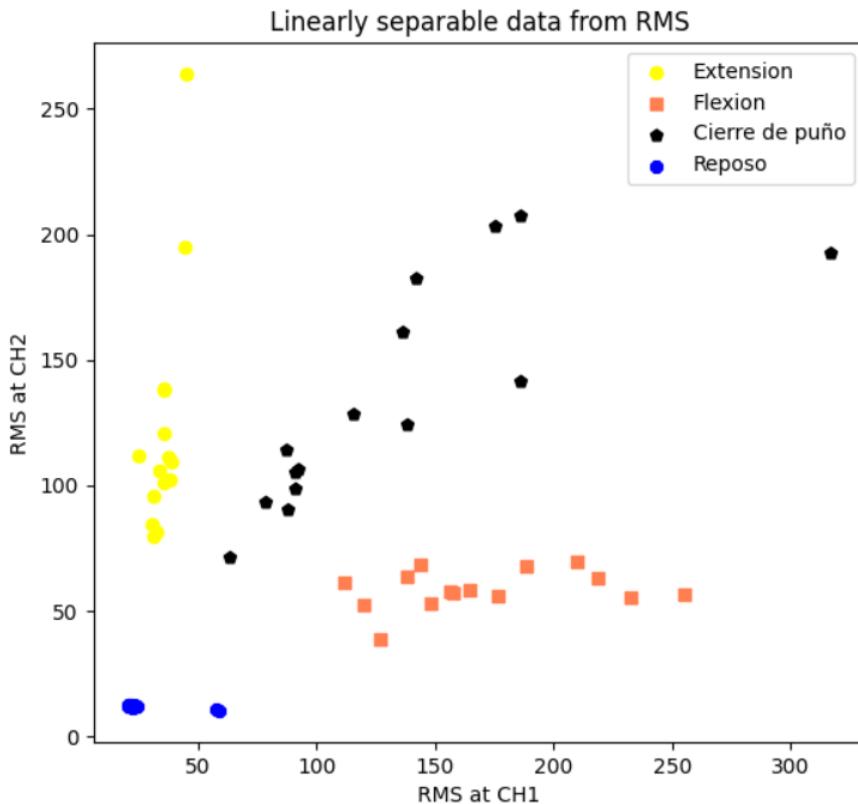


Figura 3.5: RMS para cada uno de los cuatro movimientos: reposo (azul), extensión (amarillo), cierre de puño (negro) y flexión (naranja).

En la Figura 3.5 tenemos una representación en 2 dimensiones de la RMS para los 2 canales de lectura y 15 mediciones en cada clase. El grupo de puntos azul se corresponde al reposo, negro al cierre de puño, el naranja a la extensión y azul a la flexión. Como se puede apreciar, los datos para las mediciones muestran lo que veníamos diciendo: la clasificación espacial es efectiva y los valores de RMS durante las contracciones son lo suficiente como para diferenciar bien los movimientos o grupos de puntos. Para este caso, un clasificador simple podría ser capaz de identificar fácilmente cada una de nuestras clases.

Viendo este mapa podemos ahora entender la salida de nuestro clasificador. La Figura 3.6 se corresponde a los puntos de la anterior figura y su modelo entrenado. Cada zona de color representa una clase en la que cualquier punto que caiga dentro será etiquetado como la misma. Estos resultados también son una buena opción para ver de forma gráfica que los modelos Lineales y Polinomiales son los que mejor han conseguido definir y agrupar los movimientos, mientras que los otros 2 núcleos no se ajustan para nada a la naturaleza de nuestros datos.

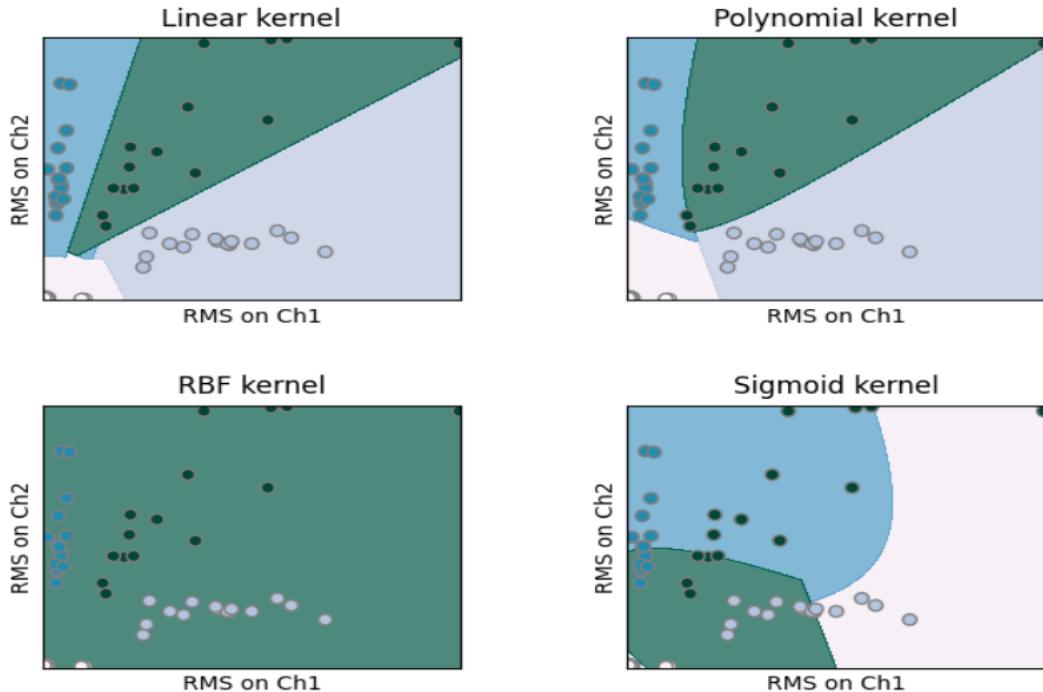


Figura 3.6: Mapa de clasificación para distintos modelos.

### 3.4. Aplicaciones

En este punto, el sistema desarrollado ya puede leer señales musculares, procesarlas y reconocer los movimientos que el usuario realiza. De esta forma, en nuestro trabajo hemos integrado dicho sistema con dos aplicaciones que pueden ser controladas mediante la actividad muscular: un videojuego sencillo y un modelo 3D de una prótesis de antebrazo que es capaz de seguir los movimientos reconocidos en tiempo real.

Para facilitar la interacción con los usuarios y mejorar la usabilidad del sistema, se desarrolló una interfaz gráfica de propósito general que permitiese acceder a sus distintas funcionalidades, incluyendo la ejecución de estas aplicaciones. Esta interfaz también permite trabajar con las señales EMG de forma intuitiva, permitiéndonos analizar los datos extraídos o las señales en bruto.

La Figura 3.7 muestra la interfaz principal, como se puede ver, empezando desde arriba a la izquierda, podemos:

- Insertar el nombre que le queremos dar al archivo de la medición. Se encargará también de llevar la cuenta de las mismas. Es decir, si se realizan varias mediciones de un mismo usuario el sistema se encarga de enumerarlas.

- El botón “MAKE MEASURE” permite hacer mediciones de hasta 2 canales. Cuenta con una simple interacción al usuario mediante un sistema de sonidos para indicar inicio de la captura de datos (veremos más detalle en el capítulo siguiente). Además, una vez realizada la medida, nos enseña los resultados y permite, en caso de error, descartarla.
- El siguiente botón “REAL TIME PLOT” dibuja las señales EMG en tiempo real.
- Al pulsar en “ANALYZE DATA”, el sistema analiza todas las mediciones que se hicieran durante la sesión para un usuario. Para ello, las guarda en una carpeta etiquetada. Además de esto último, calcula el umbral entre reposo y flexión.
- Se puede también, definir manualmente un umbral para la conexión con el juego o modelo 3D.
- Ejecutar o parar el juego o el modelo 3D al presionar los botones: “PLAY!”, “CONNECT 3D!”, “STOP!”.

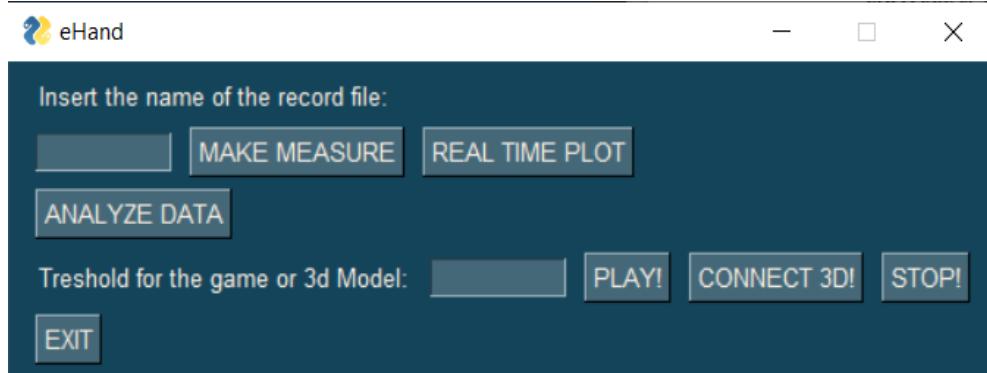


Figura 3.7: Interfaz de la aplicación.

A continuación detallaremos las dos aplicaciones que pueden ser controladas a través de nuestro sistema EMG.

### 3.4.1. Videojuego de saltos

Con el método del umbral explicado antes y nuestra aplicación, actualmente el usuario puede interactuar con un juego. La Figura 3.8 muestra un ejemplo de lo que se llega a conseguir con esta interacción. Se puede encontrar en una web [33] conocida y para los experimentos utilizaremos uno de los niveles en los que el usuario debe hacer saltar a un personaje en el momento justo para no perder.

El sistema consigue interactuar con el juego mediante la detección de la flexión en la muñeca. Simplemente, cuando esto sucede, ejecuta la acción de un “click” del ratón como si se tratase de un periférico más conectado al ordenador.

La idea nos permite, además de hacer saltar al personaje, controlar cualquier aplicación del ordenador. Esta es la versatilidad que buscamos para que la técnica del umbral o la máquina SVM pueda ser utilizada en cualquier otro juego o interfaz hombre-máquina.

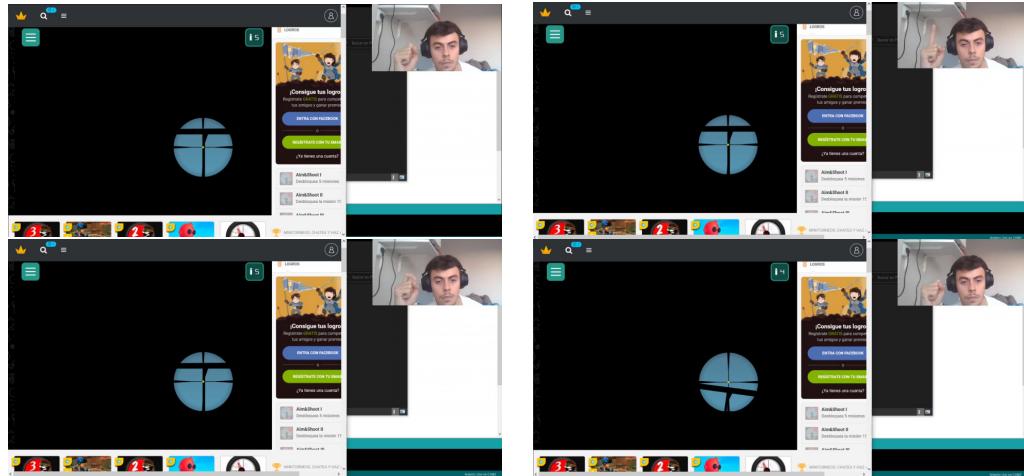


Figura 3.8: Ejemplo de EMG como interfaz para juegos.

### 3.4.2. Modelo prótesis 3D

La segunda aplicación desarrollada fue el modelo 3D de una prótesis de antebrazo controlada a través de nuestro sistema. Esta fue construida en Blender, una herramienta que permite también la ejecución de programas en Python, lo que nos permitió conectarla al EMG del usuario y seguir los movimientos de la persona en tiempo real.

El código es lo suficientemente abstracto como para ser utilizado mediante el método de umbral o la máquina SVM. En el primer caso, el modelo simplemente abrirá y cerrará la mano, mientras que en el segundo también flexionará y extenderá la muñeca. La Figura 3.9 muestra el modelo 3D de dicha prótesis abriendo y cerrando la mano.

Actualmente, tenemos 4 movimientos mapeados que podríamos pensar que se corresponden con los que debería hacer la prótesis. Este planteamiento es completamente correcto, pero entonces nos quedamos con solo 4 acciones de las muchas que tiene una mano real. Por ello, no estaría mal alejarse un poco del concepto de mimetizar al 100% cada uno de los estados que existen y utilizar los pocos recursos que tenemos para conseguir más.

Actualmente, esto ya se hace en las prótesis mediante el concepto de comandos donde se utilizan algunos de los movimientos entrenados para que funcionen a modo de comodín.

Mediante la combinación de estos y el estado actual de la máquina, se pueden codificar más de 4 acciones en nuestro sistema. Por ejemplo, si utilizamos el cierre de puño como comodín, el usuario puede ir cambiando entre diferentes posiciones de agarre. Finalmente, mediante la extensión y flexión puede regular el ángulo de cierre o apertura que le quiere dar a esa posición. Otro ejemplo, podría ser usar el cierre de puño para alternar en diferentes movimientos. En un primer estado se controlaría cierre y apertura de mano. Durante el segundo estado, se controlaría la rotación del antebrazo. La cantidad de apertura, cierre o de rotación se controlaría con la flexión y extensión de la muñeca. Como anotación, decir que este tipo de comodines también pueden existir en forma física, un simple botón en la prótesis que commuta entre diferentes las posibilidades.

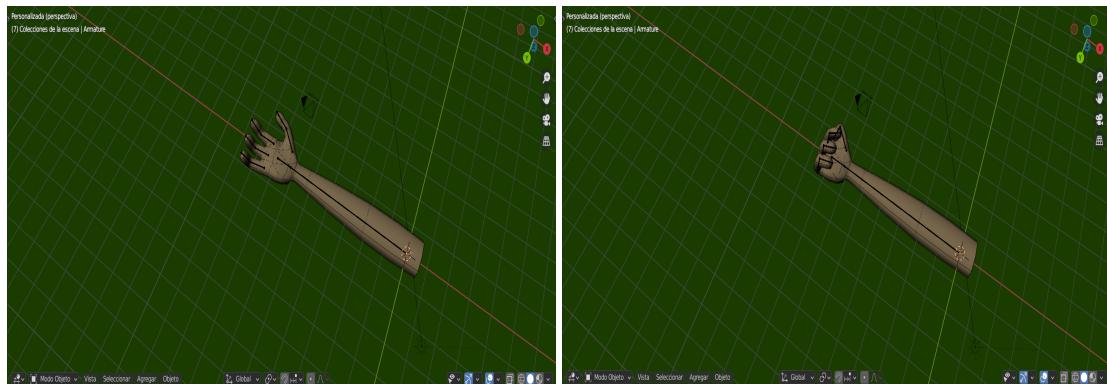


Figura 3.9: Ejemplo de cierre de puño en el modelo 3D desarrollado.

### 3.5. Código de acceso libre

Como se ha indicado anteriormente, el código desarrollado está completamente disponible en el repositorio de Github, eHand [16]. Se trata de un repositorio en continuo cambio y sujeto a mejoras, pero, por ahora, la estructura de directorios se divide principalmente en 4 partes:

- El primero será el directorio “doc” donde se encontrará toda la documentación relacionada con el proyecto.
- Seguimos con “hardware”. Aquí está todo el código que controla los microcontroladores y sensores, se encarga también de las comunicaciones con la aplicación.
- Carpeta “measures” y “dataset”. Contienen mediciones y conjuntos de datos utilizados para los análisis.
- En el siguiente directorio, “python”, está todo el código relacionado con el desarrollo de las aplicaciones. En esta estructura nos podemos encontrar:

- La carpeta “eHand” contiene la aplicación principal, interfaz gráfica y respectivas llamadas a los diferentes módulos.
- El resto de directorios se corresponden entonces a los diferentes módulos o librerías que serán utilizadas por la aplicación principal.
- Dentro de estos directorios, tenemos código para las mediciones, análisis, modelos de clasificación e incluso los demos para el juego o modelo 3D.

En primer lugar debe configurarse el Arduino utilizando el código disponible en el directorio “hardware”: “olimex\_emg\_reader.ino” se corresponde a la configuración de la [Figura 3.2](#) izquierda y “analog\_emg\_to\_serial.ino” a la de la derecha. Al ejecutarlo, solo haría faltar enviar la palabra “ini” por la herramienta Serial Plotter de Arduino para que aparezca una gráfica en tiempo real de las señales que están capturando los dispositivos.

Posteriormente, ejecutaremos el fichero “eHand.py” del directorio “python/eHand”. Esto permitirá abrir la interfaz mostrada en la [Figura 3.7](#) que, como se ha indicado anteriormente, permite ejecutar todas las funcionalidades: tomar medidas, analizar los datos, jugar y controlar el modelo 3D.

Igualmente, actualizaremos el archivo “README” en el repositorio de GITHUB con toda la información necesaria para el uso y manipulación del código.

### 3.6. Conclusiones

En este capítulo se han descrito los módulos de la arquitectura desarrollada en el TFG. El primer módulo consiste en la adquisición de señales utilizando electrodos y un sistema de conversión A/D. El segundo es la extracción de características y su posterior clasificación utilizando umbrales o máquinas de soporte vectorial. Finalmente, se ha mostrado la interfaz que permite realizar la adquisición de señales, el entrenamiento y ejecutar dos aplicaciones: juego y modelo 3D de una prótesis.



## Capítulo 4

# Pruebas con usuarios

---

**E**n este punto ya hemos conseguido desarrollar un sistema capaz de leer, procesar y aplicar señales EMG para el control de interfaces hombre-máquina. Ahora es cuando surge una de las preguntas más importantes ¿Funcionará y se obtendrán buenos resultados en usuarios que no conocen nada sobre EMG? Para responder a esta pregunta, se realizaron distintas pruebas de las dos configuraciones con usuarios ajenos al desarrollo.

Dedicamos este capítulo a describir la experiencia de probar las aplicaciones con usuarios sin ningún tipo de conocimiento o previa utilización de este tipo de tecnología. Hablaremos de ciertas prácticas que fueron cruciales de cara al éxito en los experimentos.

### 4.1. Sujetos de prueba

Contaremos con 5 personas para las pruebas (ver [Tabla 4.1](#)): una mujer de mediana edad, 2 chicos y 2 chicas jóvenes. Para los 2 últimos usuarios solo pudimos realizar las pruebas del umbral. En el resto de ellos pudimos hacer experimentos para todas las técnicas.

Ninguno de ellos son el estudiante o los directores de este TFG, por tanto desconocen el desarrollo y los resultados que se esperan alcanzar. La configuración con las placas de Olimex será la que utilicemos para estas pruebas. El montaje es mucho más limpio, permitiendo una buena impresión ante usuarios que podrían sentirse incómodos al ver el cableado e instalación que se necesita para las otras PCB. Además, será interesante ver la comparativa que surge entre este y el anterior experimento realizado durante el [capítulo 3](#). La colocación de los electrodos y configuración del software es la misma que se ha descrito en el capítulo anterior.

En la [Tabla 4.1](#) añadimos información sobre la complexión física por cada sujeto. Recordamos que durante el [capítulo 1](#), se explicaba que la captura de señales EMG superficiales puede estar condicionada a las características físicas de la persona en cuestión. Esto podía ser su edad o complexión corporal. Intentamos, por tanto, tener pruebas con varios rangos de edad y diferentes mesotipos existentes (ver [Figura 4.1](#)): endomorfo, ectomorfo y mesomorfo.

Nombre	Edad	Complejión
Usuario 1	56	ectomorfa
Usuario 2	23	mesomorfa
Usuario 3	20	ectomorfa
Usuario 4	22	endomorfa
Usuario 5	21	mesomorfa

Tabla 4.1: Información de los sujetos para las pruebas.



Figura 4.1: Mesotipos del cuerpo humano.

## 4.2. Protocolo

Antes de hablar sobre resultados, creemos importante detallar varios aspectos a tener en cuenta a la hora de trabajar con señales de EMG. En principio, se esperaba que los logros fueran parecidos a los de las pruebas preliminares, ya que la biología no engaña: durante una contracción muscular va a haber un aumento de la potencia en la señal eléctrica de nuestro cuerpo y, por ello, contábamos con la ventaja de saber cuál era el efecto que esperábamos obtener.

En la sesión, los usuarios reciben una explicación sobre el funcionamiento del sistema y sobre como se capturan las señales EMG. Todos ellos indicaron su temor a recibir algún tipo de descarga eléctrica desde los electrodos o incluso sentir algo de corriente por los mismos como si se tratase de electro-estimulación. Por lo tanto, se les explica también que el portátil con el que se registran las medidas está desconectado de la corriente y que los experimentos no les causarán ningún tipo de daño ni molestia. En caso de sentirse incómodos, pueden abandonar las pruebas en cualquier momento que lo deseen.

Una vez se les tranquiliza y explica cuál va a ser el procedimiento, es muy importante

la colocación del brazo durante las mediciones. Los flexores, extensores de la muñeca y, en general, esas zonas del antebrazo, contienen músculos que pueden estar activos sin buscarlo y afectar a las mediciones. Por lo tanto, es necesario comenzar siempre desde una posición relajada y neutra, donde cualquier rotación del antebrazo o muñeca quede fuera de la ecuación.



Figura 4.2: Colocación de los electrodos secos Olimex. Extensor en la izquierda, flexor en la derecha.

Para empezar, es recomendable primero colocar y capturar señales solo desde 1 sensor. Con el objetivo de que todos los usuarios empleen la misma configuración y tengan las mismas condiciones, en nuestros experimentos con la técnica del umbral, los electrodos siempre fueron colocados sobre el flexor del antebrazo, al igual que en las pruebas preliminares. La [Figura 4.2](#) (derecha) muestra dicha colocación. Con esto, ya se podría empezar las mediciones para probar el método del umbral y el control del videojuego. Aun así, al principio de las sesiones es muy buena práctica comenzar revisando el estado de la actividad muscular mediante el Serial Plotter del IDE Arduino o desde nuestra herramienta, lo que nos da una primera visual sobre la correcta colocación de los electrodos.

A continuación, se les pide que hagan el ejercicio de flexionar la muñeca o el ejercicio de abrir y cerrar el puño consecutivamente para, si la colocación y configuración está bien, comprobar el aumento y disminución de la amplitud en la onda entre los estados de reposo y contracción. Indistintamente, durante una flexión de muñeca o cierre del puño, esta zona muscular siempre se contrae.

En caso de no ver ninguna fluctuación en la señal o notar que los valores devueltos no se corresponden a los anteriormente vistos, posiblemente estén sucediendo alguna de las 2 siguientes situaciones:

- La colocación de los electrodos es incorrecta;
- las condiciones físicas de la zona muscular están impidiendo la correcta lectura de las señales (porcentaje alto en grasa o baja masa muscular).

En caso de ser el primer problema, recordamos que es siempre importante asegurarse de la correcta colocación de los electrodos y jugar también con la distancia de separación entre ellos. Fue por esto que decidimos usar los electrodos secos que permiten enmendar este tipo de errores fácilmente. En caso de emplear los electrodos húmedos, habría que despegarlos, desechar los parches (si se encuentran deteriorados) para coger unos nuevos y volver a probar con otro posicionamiento.

Si se trata de la segunda causa, aquí estaremos muy condicionados por el hardware que utilicemos. Con las placas de Aliexpress, no tendremos posibilidad de ajustar ningún tipo de ganancia para aumentar la sensibilidad del equipo. En cambio, con las placas Olimex tenemos la posibilidad de ajustar físicamente la ganancia del equipo. Hay que decir igualmente que en nuestro caso, no hemos sufrido esta problemática en ninguna de las pruebas tanto con una placa como con la otra.

Siguiendo con la sesión, en este punto también es normal ver, ante el desconocimiento, como los usuarios fuerzan el cierre del puño y hacen más fuerza de la necesaria, pudiendo llegar a provocar fatiga en la zona. Para solucionar este problema, se les explica que no es necesario ejercer una fuerza desmedida y que, si controlan mejor el movimiento se obtendrán también mejores resultados. Resulta más efectivo realizar movimientos rápidos de contracción, como cuando se nos cae algo de la mano y, por acto reflejo, intentamos cogerlo en el aire.

Comprobado ya que los electrodos han sido colocados correctamente y que el sistema captura adecuadamente las señales, podemos pasar ahora sí, a hacer las primeras mediciones. Recordamos entonces que con 10 grabaciones del reposo y contracción ya podemos calcular un umbral funcional para el juego. Todo se hace a través de la interfaz gráfica de nuestra aplicación eHand y, para que entiendan como va a ser el proceso, empezaremos por las lecturas del reposo, donde no tienen que interactuar de ninguna manera ni realizar ningún tipo de movimiento, pudiendo así familiarizarse con el transcurso de la toma de muestras.

La aplicación ha sido desarrollada para que durante esta fase de entrenamiento, escuchen un pitido significativo que tiene una duración de 2 segundos. Se les explica que en caso de tener que hacer algún movimiento con la mano tendría que ser en el preciso momento al dejar de escucharlo. En estas primeras mediciones, al tratarse del estado en reposo, no necesitan hacer nada.

Una vez realizadas las de reposo, pasamos a capturar las correspondientes a las de contracción/flexión de muñeca. Para ello, tal y como se ha comentado, la aplicación utilizará una alerta sonora, indicándole al usuario cuando debe realizar dicho movimiento. La flexión debe ser ejecutada justo después de que la alerta sonora termine. El sistema capturará muestras durante 1 segundo, o lo que vienen a ser 1000 muestras con la configuración que se utiliza. Recordamos que siempre es mejor tener de sobra para los análisis y este número de ellas nos pareció lo bastante significativo.

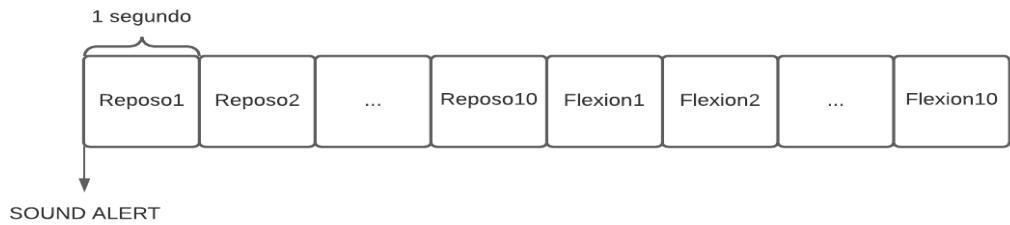


Figura 4.3: Esquema de la pauta seguida en las mediciones para la técnica del umbral.

Después de esto, mostrará por pantalla una gráfica con las señales capturadas. Mediante esta acción el estudiante puede comprobar las señales y decidir si se ha realizado una buena medición o no. Repetiríamos este proceso descrito en la Figura 4.3 hasta tener todas las capturas hechas. En general, se suele tardar entre 10 y 15 minutos, dependiendo de la calidad que se consiga durante las grabaciones y si es necesario repetir alguna de ellas.

La técnica de emitir un sonido antes de cada captura es un pequeño truco de atención para cuando tocan las mediciones de contracción o movimiento. Conseguimos que estas sean rápidas y significativas. Así, nos aseguramos obtener los mejores resultados. Finalmente, una vez hecho esto y calculado el umbral, podemos pasar a una de las aplicaciones.

### 4.3. Juego de saltos

Una vez terminada la fase de entrenamiento, se procede a probar una de las aplicaciones: el juego. La Figura 4.4 muestra una foto tomada durante la sesión.

Como se ha mencionado en el capítulo anterior, el objetivo de esta aplicación es que el usuario sea capaz de controlar un videojuego a través de la flexión de la muñeca. Consiste en hacer saltar al personaje principal para esquivar una serie de obstáculos a lo largo de un mapa. Si se consiguen saltar todos los obstáculos, se supera el nivel.

En cada experimento se le dedica tiempo hasta que la persona es capaz de superarlo sin dificultad. En menos de 10 intentos lo consiguen, suelen dedicarle entre 5 y 10 minutos.

Puede que durante las mediciones, el usuario no haya comprendido del todo la esencia del sistema y como interactuar con él, pero una vez se pone el juego a funcionar, dicha percepción cambia. La interacción directa con algo tan simple como hacer saltar a un personaje con el mero hecho de mover la mano les hace entender mejor el funcionamiento del sistema e, incluso, les ayuda a dominar este tipo de interfaz.

Con el objetivo de comparar los resultados obtenidos con los usuarios, la Figura 4.5 se muestran los valores de RMS calculados durante la fase de entrenamiento para tres usuarios.



Figura 4.4: Usuario probando el juego.

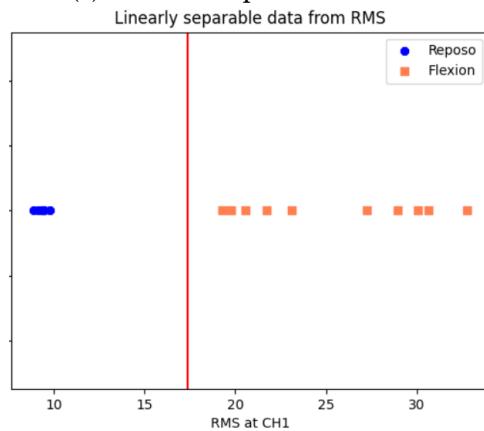
También se muestra el umbral obtenido (línea roja). Los valores de flexión (naranja), son lo suficientemente significativos como para poder realizar la clasificación. Es más, podemos ver que las mediciones en reposo (azul), se mantienen estables en cuanto a RMS y forman una nube de puntos muy consistente, permitiendo así calcular un umbral (línea roja) que separe correctamente las 2 zonas de movimiento. En la [Tabla 4.2](#) se muestra el valor calculado para el umbral de cada uno de los usuarios. Como puede verse, hay gran diferencia entre algunos umbrales, lo que indica la importancia de calcular el valor para cada usuario en vez de emplear uno prefijado.

El juego consta de 7 saltos y la dificultad radica en que los usuarios deben realizar la acción en el momento correcto para conseguir pasarse el nivel. En general, los resultados durante esta prueba fueron satisfactorios en todos los casos: los que lo probaron consiguen después de algunos intentos pasarse el nivel sin mayor problema. La [Tabla 4.2](#) muestra el número de intentos que tuvieron que realizar para conseguir el objetivo.

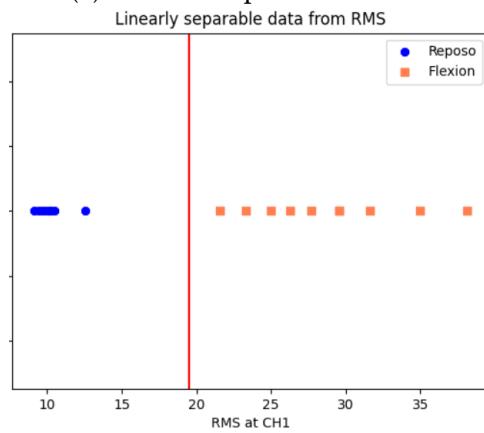
Juego	Umbral	Intentos
Usuario 1	17.3	5
Usuario 2	19.4	3
Usuario 3	25	4
Usuario 4	4.5	6
Usuario 5	4.4	4

Tabla 4.2: Umbral y número de intentos necesarios para superar un nivel del juego por cada usuario.

(a) Resultados para el usuario 1



(b) Resultados para el usuario 2



(c) Resultados para el usuario 4

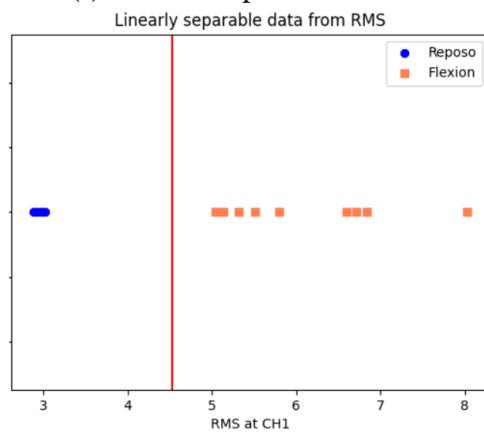


Figura 4.5: Resultados RMS (en orden) para los distintos usuarios: reposo (azul), flexión (naranja), linea roja el umbral calculado.

#### 4.4. Modelo 3D

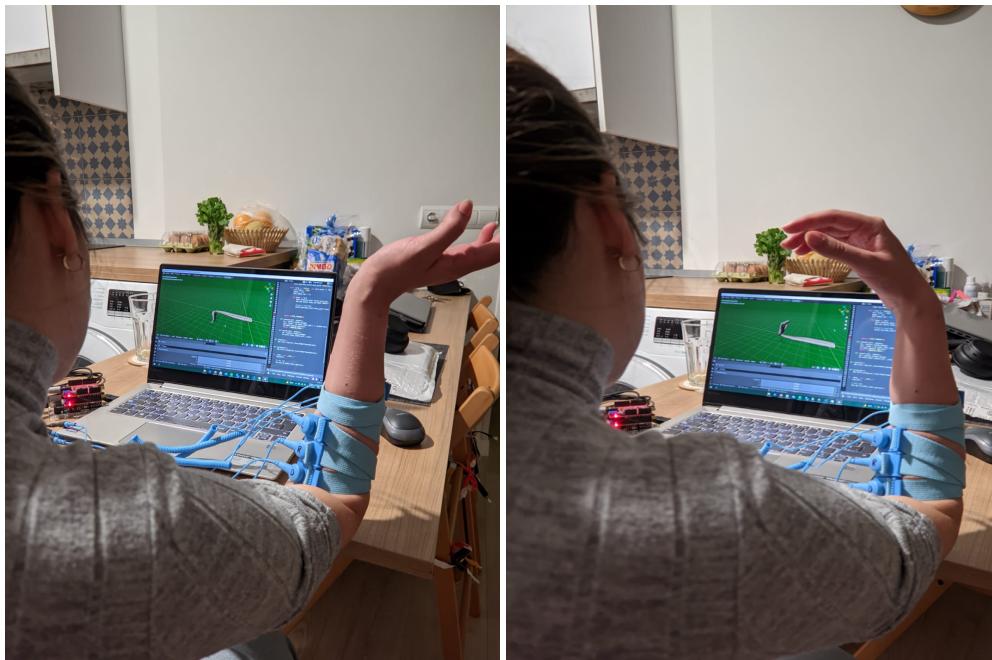


Figura 4.6: Usuario probando el modelo 3D.

Habiendo tenido éxito con el juego, se puede probar el modelo 3D (ver Figura 4.6) con el método del umbral o, pasar directamente ya a colocar el segundo sensor para comenzar con las mediciones de las otras clases de movimiento necesarias para entrenar y construir el modelo SVM.

Como la persona ya conoce el procedimiento y entiende mejor la situación, este proceso se haría de forma rápida. Recordamos que para el correcto entrenamiento de la máquina, usaremos 15 mediciones de cada clase. El experimento tendría una duración aproximada de unos 20 minutos.

Como ejemplo, la Figura 4.8 muestra la pantalla en la aplicación de Blender, donde se pueden apreciar todas las clases de movimiento que se han conseguido caracterizar en este trabajo. Tenemos, empezando desde arriba a la izquierda, el reposo, puño cerrado, flexión y extensión de muñeca.

Esta sección se va a dedicar principalmente a los resultados que se consiguieron durante el entrenamiento y validación de la máquina SVM para los 3 primeros usuarios. Esta parte es importante de cara a un futuro poder implementar una prótesis con la inteligencia suficiente para reconocer varios movimientos.

A modo de ejemplo, la Figura 4.7 muestra los mapas obtenidos para los usuarios 1 y 2 con cuatro núcleos diferentes. Como puede verse, los núcleos lineal y polinomial realizan una

mejor separación de los estados para los dos usuarios.

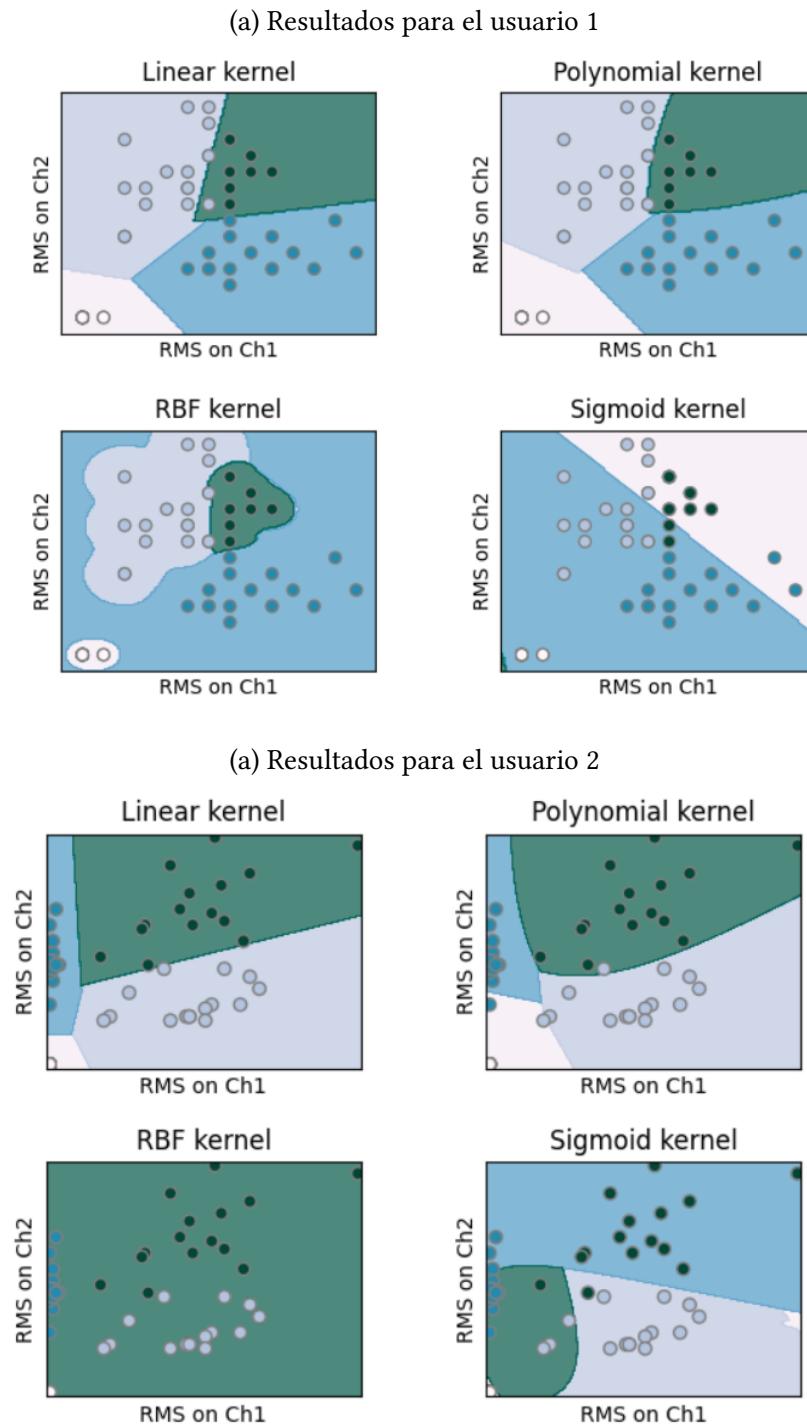


Figura 4.7: Mapas de clasificación para los usuarios 1 y 2.

Para validar el correcto funcionamiento de la máquina SVM, se utilizó k-fold como método estadístico para estimar la habilidad de clasificación que consiguen con los datos obtenidos en las mediciones. K-fold [32] consiste en validar cruzadamente un modelo, dividiendo y “plegando” los conjuntos de datos en distintos niveles que servirán para entrenar y probar  $k$  modelos de forma mezclada. Se trata de un procedimiento de remuestreo que se utiliza para evaluar modelos de aprendizaje automático con una muestra de datos limitada.

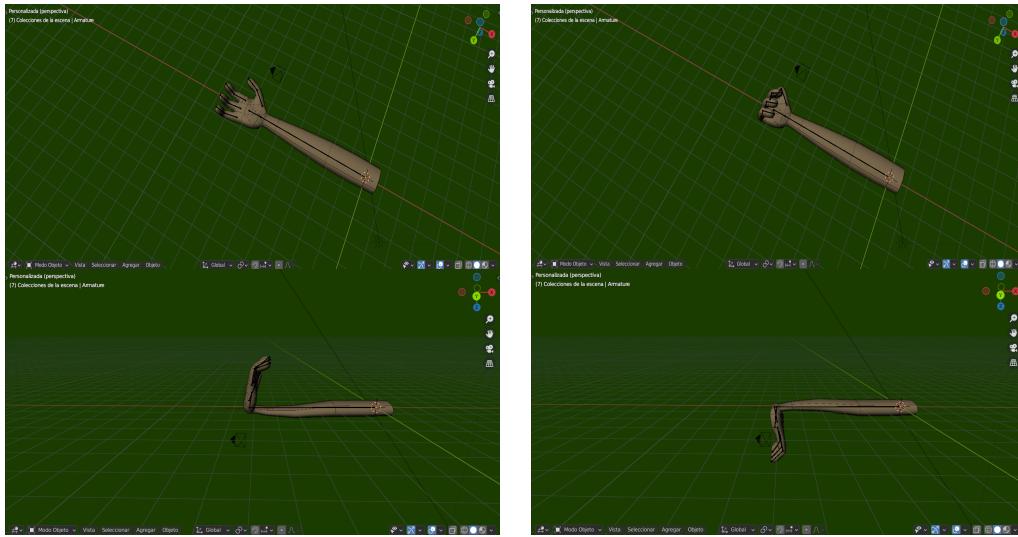


Figura 4.8: Ejemplo de todos los movimientos conseguidos en el modelo 3D.

Como tal, el procedimiento a menudo se denomina validación cruzada de  $k$  veces. Por ejemplo, cuando  $k = 10$  esta referencia se convierte en una validación cruzada de 10 veces. En nuestro caso y por el tamaño de nuestros datos, durante todos los experimentos utilizaremos una validación cruzada de 3 veces.

Este método se utiliza principalmente para estimar la habilidad de un modelo con datos desconocidos. Es decir, utilizamos una muestra limitada de datos para estimar cómo se espera que funcione el modelo en general cuando se utiliza para hacer predicciones sobre otro conjunto que no es usado durante el entrenamiento. El algoritmo es popular por su sencillez y por proporcionar resultados poco sesgados, ya que se encarga de entrenar y evaluar el modelo con todos los datos disponibles. El procedimiento general de k-fold es el siguiente:

- Mezclamos el conjunto de datos de forma aleatoria.
- Dividimos el conjunto de datos en  $k$  grupos.
- Cogemos uno de los grupos como conjunto de datos de prueba.
- Tomamos los grupos restantes como un conjunto de datos de entrenamiento.

- Entrenamos y evaluamos.
- Repetimos este proceso hasta haber evaluado el modelo con cada 1 de los grupos únicos que se formaron al principio.
- Extraemos la habilidad del modelo usando las puntuaciones obtenidas en cada iteración.

Finalmente, una vez validados los diferentes modelos mediante k-fold, podremos decidir, en base a la puntuación, cuál será el que utilicemos en el desarrollo final de nuestro sistema. El modelo que mejores resultados de precisión obtenga, será entrenado con todo el conjunto de datos disponible. Este modelo será el que se utilice para probar la aplicación del modelo 3D en usuarios finales.

Los resultados para la validación de k-fold por cada usuario y núcleo, se encuentran en la [Tabla 4.3](#). En todos los resultados, el valor de acierto se representa mediante un número real contenido entre 0 y 1, entendiéndose los dos como mínimo y máximo de eficacia.

Usuario	Núcleo Lineal	Núcleo Polinomial	Núcleo Radial	Núcleo Sigmoide
Usuario 1	0.935	0.903	0.805	0.098
Usuario 2	0.983	0.950	0.400	0.067
Usuario 3	0.933	0.883	0.567	0.400

Tabla 4.3: Resultados para 3-fold en los 4 kernels y por cada usuario.

En vista de los resultados arrojados por k-fold, los experimentos son muy satisfactorios para el núcleo lineal. Elegiremos entonces a este ganador por ser, además, más eficiente computacionalmente que el resto.

Al probarlo en tiempo real, una vez entrenado el modelo con todo el conjunto de datos disponible, se puede ver como la máquina SVM cumple perfectamente con los requisitos de clasificación. Añadir finalmente, que una de las causas de este éxito también radica en el anterior entrenamiento por parte del usuario, es decir, no solo estamos configurando el modelo para que responda a cierto conjunto de datos, sino que también estamos educando al cuerpo y a la persona para que se mantenga en una cierta naturaleza de movimientos que permite mantener el acierto del sistema.

## 4.5. Conclusiones

Durante este capítulo hemos querido hacer énfasis en la interacción de las aplicaciones con usuarios reales. El estudiante se dio cuenta de que lo más difícil de cara al éxito en las pruebas no es la calidad del software ni del hardware, sino que es necesario tener el conocimiento para saber conducir las sesiones. Reaccionar debidamente a los detalles que marcan la diferencia

durante los experimentos fue algo aprendido a base de la experiencia en realizar pruebas sobre sí mismo. Esto es algo que no se encuentra en ningún libro o fuente de información, pero, a su vez, es muy importante.

Finalmente, se pudo comprobar que tanto con la configuración con la placa de Olimex o la configuración con la placa de Aliexpress se pueden conseguir muy buenos resultados. En cuanto a resultados y diferencias fisiológicas, si que vimos que para usuarios de tipo ectomorfo el sensor es capaz de reconocer mejor las señales musculares. Los valores de umbral se distinguen mejor y existe más espacio entre clases de movimientos. Para algunos mesomorfos o endomorfos, la potencia de la contracción leída disminuía. En todos los casos no significó un problema por la capacidad que tiene esta placa de ajustar su sensibilidad.

Sería interesante probar con más rangos de edad para ver si existe diferencia. Igualmente, creemos fielmente que los métodos se podrían aplicar en cualquier usuario con pequeñas correcciones.

## Capítulo 5

# Metodología, planificación y costes del desarrollo

---

En este capítulo detallaremos los conceptos aplicados a la metodología seguida para el desarrollo de las aplicaciones de las que hablamos durante este TFG y como se ajustó a la realidad del mismo. Además, se proporcionará una relación de costes y tiempo dedicado a todo el proceso.

### 5.1. Metodología

Este proyecto se basó en la metodología de prototipado rápido. Para nosotros fue difícil utilizar algún otro método donde se definieran una serie de iteraciones u objetivos, ya que dependíamos completamente de las herramientas hardware a las que fuimos teniendo acceso y de las necesidades que iban surgiendo a lo largo del desarrollo. Por lo tanto, las herramientas software creadas fueron evolucionando alrededor de un ecosistema cambiante que fue tomando forma y definiéndose a lo largo del progreso en el trabajo.

Igualmente, podemos hacer un sumario de las fases que se siguieron y del tiempo dedicado a cada una.

- Fase 1: Adquisición y prueba de distintos dispositivos hardware.
- Fase 2: Montaje de la configuración y comprobación de su correcto funcionamiento midiendo las primeras señales.
- Fase 3: Creación del dataset que posteriormente será analizado con el software.
- Fase 4: Creación del software totalmente OpenSource.
- Fase 5: Documentación de todo el proceso.

- Fase 6: Difusión de todo el proceso para que en el futuro sea útil para cualquier persona interesada en el tema.

### 5.1.1. Fase 1 y 2

Las fases 1 y 2 fueron las que más tiempo comprendieron por la dificultad que existe actualmente para conseguir información sobre EMG, sobre la adquisición de estas señales y finalmente para la compra del equipamiento. Cada placa tiene unas ciertas características que implicaron horas de análisis y búsqueda de documentación que nos permitiese entender el funcionamiento que tienen y uso que hay que darle para su posterior aplicación en nuestro desarrollo. Para el estudio de la placa OpenBCI, fue de gran utilidad la experiencia previa que tenían los directores del TFG.

### 5.1.2. Fase 3 y 4

Una vez configurado nuestro equipo de mediciones, hizo falta crear software para las mismas y manipulación de los dataset. Es por ellos que estas dos fases se realizaron en paralelo. Se trabajó en las herramientas software según iban surgiendo necesidades y se consolidaban las líneas de conducta que mejor resultados ofrecían durante los experimentos. Primero se comenzó sin ningún tipo de interfaz gráfica, creando los primeros programas de análisis y mediciones por consola. Una vez testeados y desarrolladas las técnicas de umbral y SVM surgió la idea de juntar todo en una interfaz gráfica. En esta fase, fue de gran utilidad los comentarios realizados por una terapeuta ocupacional sobre las necesidades que tienen los usuarios reales.

Igualmente, durante estas fases ya podríamos también incluir el tiempo dedicado a las pruebas del sistema sobre usuarios, ya que parte del software estaba creado y testeado.

### 5.1.3. Fase 5 y 6

Las últimas dos fases abarcan el final de este TFG, en el que, una vez aprendido de toda esta experiencia, el alumno volcó todos los conocimientos conseguidos durante estas semanas de trabajo y experimentación. Priorizó aquellos aspectos que le pueden ser útiles a cualquier persona interesada en el proyecto o que a él le habría gustado saber cuando comenzó con esto. El alumno redactó la memoria que fue revisada diariamente por los directores del TFG.

## 5.2. Planificación

La Figura 5.1 muestra la planificación realizada para el desarrollo de las tareas descritas anteriormente.

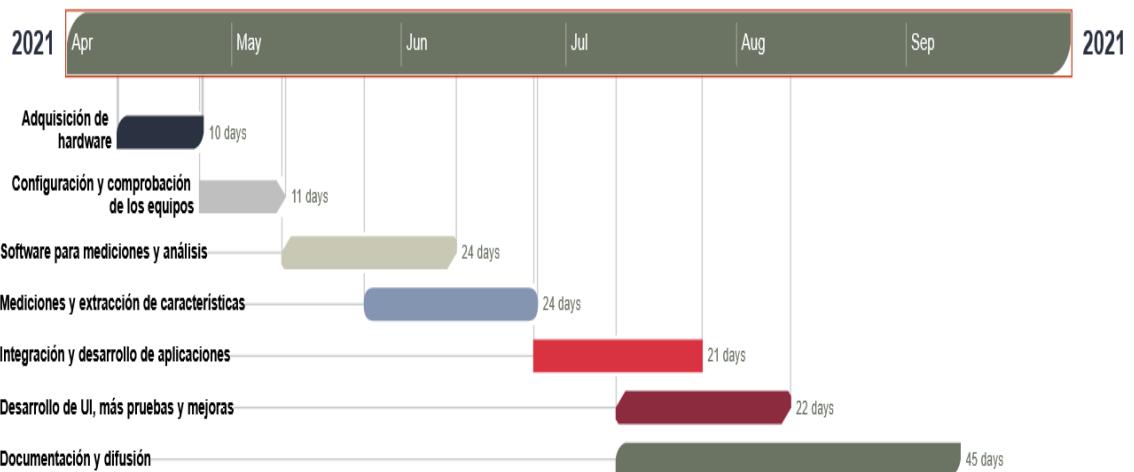


Figura 5.1: Diagrama de Gantt de la planificación inicial del proyecto.

La planificación que se pensó en un principio estaba compuesta por 5 meses, comenzando en el mes de abril y terminando en el mes de septiembre de 2021. Esta estimación sería entonces de alrededor de 5 horas diarias durante 5 días a la semana, lo que resulta en alrededor también de 25 horas/persona semanales. Si multiplicamos por las 20 semanas de trabajo, implica que se habían planificado un total de 500 horas/persona.

Fases	Fecha Inicio	Fecha Fin	Horas planificadas
1 y 2	10/04/21	10/05/21	100
3 y 4	10/05/21	10/08/21	240
5 y 6	10/07/21	10/09/21	160

Tabla 5.1: Planificación inicial para el proyecto.

En la Tabla 5.1, se encuentra un resumen de la planificación inicial que teníamos prevista para este proyecto.

### 5.3. Incidencias

A lo largo del proyecto hemos intentado seguir la planificación de las fases lo más estrictamente posible, pero debido a situaciones imprevistas y la necesidad de más pruebas con usuarios, surgieron retrasos en la realización y finalización del proyecto.

Las incidencias más importantes fueron:

- Debido a la dificultad comentada para la adquisición del hardware, la compra y envío del mismo llevó más tiempo de lo esperado. Incluso una vez recibido, hizo falta dedicar

recursos al estudio de la utilización del mismo. Para las placas de Aliexpress, el vendedor no indicaba en ninguna parte cual era su correcta configuración y por ello hizo falta, dedicar tiempo y pruebas hasta que encontramos la información necesaria para su uso. Con las de Olimex sucedió más de lo mismo. En los capítulos de introducción comentábamos que para conseguir un correcto funcionamiento se tuvo que adaptar una librería de filtrado en C++ a esta placa en cuestión que conseguía los mejores resultados. Debido a esta incidencia, las fases 1, 2 y 3 se vieron retrasadas 60 días sobre la planificación inicial.

- Las primeras mediciones realizadas con el equipamiento no ofrecían el éxito que buscábamos. Todo esto a causa de no saber y no tener un procedimiento claro a la hora de tomar las muestras. Esto mismo lo comentábamos durante el capítulo 4, la posición del brazo durante las sesiones, la colocación de los electrodos, el correcto funcionamiento del software y hardware. En un ambiente en el que había tantos factores que podían estar fallando a la vez, se consumió tiempo y recursos en marcar las pautas que nos permitiesen sacar provecho al material. Debido a esta incidencia, las fases 3 y 4 se vieron retrasadas 30 días sobre la planificación inicial.
- Para el correcto funcionamiento de la aplicación en el modelo 3D, hizo falta dedicar tiempo para entender la integración existente entre Blender y Python, a la vez que idear un mecanismo para conectar las lecturas EMG con el programa. No se trata de algo complejo. Sin embargo se necesitó bastante tiempo para entender el funcionamiento de esta integración. Debido a esta incidencia, las fases 3 y 4 se vieron retrasadas unos 15 días sobre la planificación inicial.
- Como se ha indicado, las pruebas fueron realizadas con usuarios ajenos a este proyecto porque el objetivo es probarlo con un paciente amputado. No ha sido fácil conseguir un usuario de este tipo, pero a través de la asociación COGAMI, esperamos hacer estas pruebas pronto. Por lo expuesto con anterioridad y juntando los retrasos acumulados, las fases 5 y 6 sufrieron un retraso de 60 días.

Fases	Fecha Inicio	Fecha Fin	Horas reales
1 y 2	10/04/21	10/07/21	140
3 y 4	10/05/21	10/10/21	260
5 y 6	01/09/21	10/11/21	160

Tabla 5.2: Resumen temporal del proyecto.

En la [Tabla 5.2](#) se puede ver un desglose temporal de lo que llevaría finalmente el proyecto. Como se puede ver las fases 3 y 4 se superponen sobre las otras por la necesidad que hubo de

ir desarrollando las herramientas a medida que iba llegando el material, se probaba, surgían nuevas necesidades y se desarrollaban los primeros prototipos. Finalmente, el proyecto fue desarrollado durante un total de 560 horas.

#### 5.4. Estimación de costes

La estimación de costes en este proyecto se centran en la parte hardware y costes humanos. Todo el software utilizado es Open Source y, por lo tanto, no forma parte de un coste real. La Tabla 5.3 presenta la lista de elementos que fueron necesarios para el proyecto. No se incluye la placa OpenBCI porque solo se utilizó para realizar una prueba inicial y fue descartada.

Hardware	Precio
Portátil Lenovo 530S-14IKB	58€
Arduino Uno	23€
Olimex Shield-ekg-emg x2	48,4€
Electrodos en seco Olimex	24€
PCB Aliexpress + Electrodos x2	30€
Total	<b>183,4€</b>

Tabla 5.3: Resumen de costes del proyecto.

Para el coste del portátil se ha considerado una vida útil de 5 años y un uso de 7 meses en este proyecto. El resto de material se imputa por completo al proyecto.

Por último, para el cálculo del coste de los recursos humanos, diferenciamos entre 2 figuras: desarrollador y directores del proyecto. Para el primero se consideró un coste a razón de 20€/hora y para el segundo de 30€/hora. Para las 560 horas de trabajo calculamos lo que se muestra en la tabla 5.4.

Puesto	Coste	Horas	Coste
Desarrollador	20€/hora	560	11.200 €
Director de proyecto	30€/hora	100	3.000 €
Total	-	-	<b>14.200 €</b>

Tabla 5.4: Costes de recursos humanos.

El desarrollador dedicó todo el tiempo disponible a las diferentes fases donde recibió apoyo y supervisión en todas las tareas por parte de los directores.

Finalmente, sumando todos los costes mencionados hasta el momento, el proyecto tiene un coste estimado de 14.383,4€.



## Capítulo 6

# Conclusiones y líneas futuras

---

En este último capítulo, queremos hacer un análisis de lo conseguido hasta ahora en el proyecto y de las inquietudes que le han surgido al alumno en el proceso para, finalmente, hablar del futuro que pretende seguir eHand.

### 6.1. Conclusiones

Este proyecto nació con la finalidad de construir una prótesis de mano barata y accesible para cualquiera. Parece que durante este trabajo se habla poco de esto mismo, pero en realidad el objetivo evolucionó para adaptarse al verdadero problema que existe alrededor de la construcción de un sistema de este tipo. Dicho problema reside en que la tecnología no se encuentra al alcance del individuo medio. Está reservada para un nicho que no representa al grueso de la población. Tampoco se trata de una tecnología accesible para personas con conocimientos clásicos en el área de la informática o electrónica. Por eso, con este proyecto hemos conseguido hacer una revisión de buenas alternativas hardware a bajo precio y se han creado herramientas software que ayuden a esta causa. Hemos planteado diferentes usos que se le podría dar al EMG y creemos firmemente que este es el camino a seguir. Se ha demostrado durante los experimentos que es fácil conseguir buenos resultados incluso con gente que nunca había tenido contacto con un sistema de este estilo.

Los logros conseguidos durante este TFG se resumen en:

- Hemos revisado y utilizado material de bajo coste para todos los experimentos.
- Se ha creado una plataforma software en la que poder trabajar con señales EMG de forma cómoda y rápida. Además, dicha plataforma y todo el código desarrollado para el análisis de las señales EMG han sido publicadas en un repositorio de acceso público [16], para que cualquier persona interesada pueda emplearlo.
- Se ha conseguido caracterizar movimientos de la mano tanto con 1 sensor como con 2.

- Hemos logrado utilizar estas características para controlar interfaces hombre-máquina.
- Se le ha dado un buen propósito a esto mismo. Tanto para el juego como para el control de la prótesis 3D.
- La documentación realizada recoge los detalles más importantes e información necesaria para ayudar al desenvolvimiento de la tecnología EMG.

No todo salió como esperábamos durante la realización de este trabajo, por lo que debemos también hacer un análisis exhaustivo sobre su realización para extraer lecciones y experiencias que nos ayuden de cara al futuro. Entre las lecciones obtenidas podemos destacar:

- Imprevistos: es normal que durante un desarrollo surjan imprevistos, pero hemos aprendido que es necesario intentar curarse de los mismos a tiempo. La flexibilidad de un proyecto es necesaria para evitar retrasos que se podrían compensar con la realización de otras tareas en su lugar.
- La planificación debería ser una parte esencial en todo proyecto. Se trata también de una herramienta útil para evitar problemas con el punto anterior. Durante este proyecto hubo mucho margen de mejora en este aspecto. Sin embargo, la naturaleza de un trabajo como este necesitaba tiempo y maduración para coger el cauce que tiene actualmente.
- Finalmente, la falta de experiencia jugó un papel importante durante la realización del trabajo. Durante el desarrollo y experimentación se podría haber sido mucho más metódico para evitar resultados confusos o la repetición de experimentos por falta de rigor en los que ya se habían realizado. Además, se podrían haber evitado ciertas prácticas como la de programar primero y luego diseñar. Puede que tenga sus ventajas en una metodología de prototipado rápido, pero no es razón para descuidarse. Por último, destacó también la falta de experiencia a la hora de realizar documentación técnica como es este presente documento.

## **6.2. Líneas futuras**

Creemos que este proyecto no está para nada acabado y, por ello, durante esta sección hablaremos del camino que seguiremos para su desarrollo.

Las pruebas realizadas fueron con personas que tienen la mano porque no contábamos con la colaboración de usuarios sin este miembro. A través de la asociación COGAMI, pensamos que próximamente vamos a poner realizar pruebas con un paciente real. Sin embargo, esto requiere realizar un estudio previo sobre varios aspectos como, por ejemplo, tipo de amputación, fuerza de los músculos en la zona, etc.

Respecto a la aplicación que se enseñó durante este trabajo, la idea sería que evolucionase en una plataforma Open Source donde cualquier interesado pueda desarrollar cualquier tipo de herramienta. No solo con fines médicos. Se contempla su uso para la realidad virtual o cualquier otro tipo de interfaz hombre-máquina controlada por señales musculares.

Sería interesante que la aplicación consiguiese ser capaz de:

- Permitir al usuario conectarse con el Arduino o el microcontrolador que utilice de forma inalámbrica. Hemos pensado en utilizar la conexión bluetooth para este propósito.
- Actualmente la aplicación solo es capaz de procesar hasta 2 canales de señales EMG. Queremos en un futuro aumentar este número hasta 4 u 8.
- El programa encargado del entrenamiento y uso de la máquina SVM actualmente no se encuentra integrado en la aplicación. Esto también será solucionado.
- Comentábamos en capítulos anteriores que el truco para controlar la aplicación del videojuego residía en mapear uno de los movimientos como si se tratase el “click” de un ratón. Bien, en siguientes iteraciones queremos permitir al usuario elegir cual quiere que sea el botón o la acción que se ejecute con la caracterización del movimiento. Esto le permitirá probar el sistema en diferentes ámbitos que no solo sean el juego de saltos que actualmente tenemos.
- Dar más posibilidades de análisis, no solo RMS. Sería interesante que la aplicación fuera capaz de jugar con algún método estadístico más. Incluso la opción de poder aplicar transformaciones en frecuencia o filtros también ayudaría a los futuros usuarios durante sus investigaciones.
- Igualmente, sería entonces comprensible hacer un análisis sobre el co-diseño hardware y software para buscar un equilibrio que nos beneficie a todos. Por ejemplo, en este TFG estudiaremos modelos de vectores de soporte, pero estos no son los únicos. No son los únicos, pero si cumplen la restricción de sencillez y rapidez en la toma de decisiones. También podría suceder que otro algoritmo de clasificación como las Redes de Neuronas o modelo de Bayes combinen mejor con la arquitectura. Además de esto, hay muchas otras cualidades de la mano que podrían ser mimetizadas y que quizás impliquen una serie de mecanismos software que combinan mejor con un hardware u otro. Por ejemplo, durante el proyecto pudimos clasificar movimientos de la mano, pero, ¿cómo se podría caracterizar la velocidad o momento angular e incluso la fuerza a la que se debería producir el movimiento en base a señales EMG?
- Finalmente, se quiere hacer correcciones en la interfaz gráfica que existe ahora mismo, hacerla más atractiva de cara a los usuarios y mejorar la arquitectura software que existe

por detrás.

Esto sería lo que tenemos pensado para las herramientas software. ¿Qué pasará con el hardware? En principio queremos hacer 2 cosas: construir un electromiógrafo con las PCB de Aliexpress y probar alguna de las placas descritas durante el capítulo 3. Para el primer propósito usaremos las siguientes herramientas:

- Queremos que el electromiógrafo disponga de conexión bluetooth. Y por ello tenemos pensado utilizar como elemento central al microcontrolador ESP32. Como se puede ver en la Figura 6.1 dispone además de 2 conversores ADC y se puede alimentar con una entrada de entre +3.3 V y +5 V. También existe la alternativa de utilizar un Arduino Nano conectado a una placa de expansión bluetooth. Por ahora nos decantamos por la primera opción que contiene todo en un mismo pack.
- Para la alimentación de las placas contaremos con una batería recargable. Necesitaría de un circuito para la carga y circuitería para regular el voltaje de salida. Todos estos componentes son baratos y fáciles de conseguir.

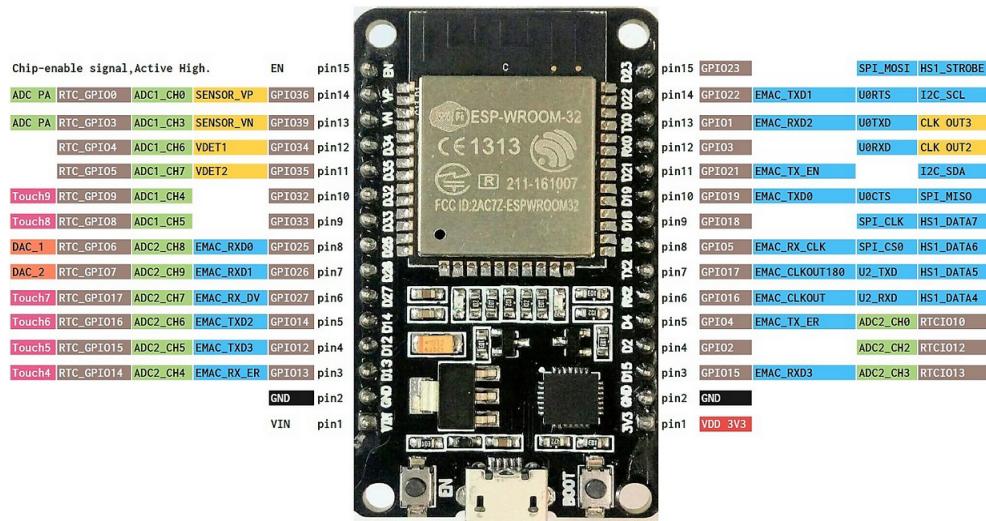


Figura 6.1: Microcontrolador ESP32, con tecnología WiFi y Bluetooth.

Respecto al segundo objetivo (seguir probando sensores) tenemos pensado adquirir la placa de Myoware por las cualidades que ofrece en diseño, recolección de señales y configurabilidad. Por otra parte, compraremos nuevos electrodos para intentar amoldarlos al uso con la placa OpenBCI disponible en la Universidad, será interesante ver el alcance de los análisis que se consiguen gracias a la experiencia que hay con esta PCB. En relación a las placas de

## CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS

---

Olimex, pensamos que la actual librería de filtrado que se utiliza sobre la misma no le saca todo el partido que buscábamos y por lo tanto, sería interesante echarle un vistazo a los filtros que utiliza para mejorar sus resultados.

Por último, y aunque hay mucho margen de mejora en las aplicaciones desarrolladas, el producto existente ya nos permite pasar a pensar en la construcción de un prototipo para una prótesis de mano. El paso previo de fabricación para el electromiógrafo nos ayudará a comprender las necesidades que existen a la hora de manejar cierta circuitería y que también existen en el modelado e impresión 3D.



# Bibliografía

---

- [1] “Open bionics,” <https://openbionics.com/en/hero-arm/>, [accessed 10-November-2021].
- [2] G. Arnold, L. Ziaoou, G. Fortino, R. Perez, V. Lumelsky, L. Shafer, and D. Goldgof, *Surface Electromiography: Physiology, Engineering and Applications*. John Wiley & Sons Inc., Hoboken, New Jersey, 2016.
- [3] MDURANCE, “¿Qué es la electromiografía de superficie?” <https://blog.mdurance.eu/academia/que-es-la-electromiografia-de-superficie/>, 2019.
- [4] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, “Application of wavelet analysis in emg feature extraction for pattern classification,” 2011.
- [5] “Olimex,” <https://www.olimex.com/Products/Duino/Shields/SCHILD-EKG-EMG/open-source-hardware>, [accessed 03-September-2021].
- [6] “Aliexpress,” <https://es.aliexpress.com/>, [accessed 10-September-2021].
- [7] “Advancer technologies, myoware,” <http://www.advancertechnologies.com/p/myoware.html>, [accessed 5-November-2021].
- [8] “Oymotion, gravity,” <https://www.dfrobot.com/product-1661.html>, [accessed 5-November-2021].
- [9] “Openbci,” <https://shop.openbci.com/products/cyton-biosensing-board-8-channel?variant=38958638542>, [accessed 20-September-2021].
- [10] “Mdurance,” <https://mdurance.eu/>, [accessed 16-September-2021].
- [11] J. Ribeiro, F. Mota, T. Cavalcante, I. Nogueira, V. Gondim, V. Albuquerque, and A. Alexandria, “Analysis of man-machine interfaces in upper-limb prosthesis: A review,” *Robotics*, vol. 8, no. 1, p. 16, 2019.

- [12] J. R. Cram, *CRAM'S INTRODUCTION TO: Surface Electromyography*, 2nd ed. Jones and Bartlett, 1998.
- [13] “Ossur, soluciones protésicas,” <https://www.ossur.com/es-es>, [accessed 25-October-2021].
- [14] “Biosignalsplex,” <https://biosignalsplex.com/products/sensors/electromyography.html>, [accessed 15-October-2021].
- [15] “Canal de biomakers inc.” <https://www.youtube.com/c/BioMakers>, [accessed 25-October-2021].
- [16] “Repositorio ehand,” <https://github.com/aleir97/eHand>, [accessed 10-November-2021].
- [17] L. van Dijk, C. van der Sluis, H. van Dijk, and R. Bongers, *Learning an EMG Controlled Game: Task-Specific Adaptations and Transfer*. PLoS One., 2016.
- [18] “Web oficial de biomakers inc.” <https://biomakers.tech/>, [accessed 15-October-2021].
- [19] “Altbionics,” <https://altbionics.com/>, [accessed 15-October-2021].
- [20] “Youbionic,” <https://www.youbionic.com/>, [accessed 15-October-2021].
- [21] “Asociación cogami,” [https://www.cogami.gal/es\\_ES/](https://www.cogami.gal/es_ES/), [accessed 5-October-2021].
- [22] “Prótesis michelangelo,” <https://www.ottobock.es/protesica/miembro-superior/sistemas-de-brazo-y-mano/axon-bus-con-mano-michelangelo/>, [accessed 16-September-2021].
- [23] Arduino, “Documentation of the Arduino Uno R3,” <https://www.arduino.cc/>, [accessed 25-October-2021].
- [24] Python, “Documentation for Python,” <https://docs.python.org/>, [accessed 25-October-2021].
- [25] Matplotlib, “Main webpage of the Matplotlib python library,” <https://matplotlib.org/>, [accessed 25-October-2021].
- [26] Numpy, “Main webpage of the Numpy python library,” <https://numpy.org/about/>, [accessed 25-October-2021].
- [27] PySerial, “Main webpage of the PySerial python library,” <https://pyserial.readthedocs.io/en/latest/index.html#>, [accessed 25-October-2021].

## BIBLIOGRAFÍA

---

- [28] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
- [29] “Blender,” <https://www.blender.org>, [accessed 25-October-2021].
- [30] L. Myers, M. Lowery, M. O’Malley, C. Vaughan, C. Heneghan, A. St Clair Gibson, Y. Harley, and R. Sreenivasan, “Rectification and non-linear pre-processing of emg signals for cortico-muscular analysis,” *Journal of Neuroscience Methods*, vol. 124, no. 2, pp. 157–165, 2003. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0165027003000049>
- [31] A. M. Sanz and L. M. Lorente, *Procesado Avanzado de Señal EMG*. Universidad Carlos III de Madrid, 2017.
- [32] Scikit-learn, “Documentation from Scikit-learn,” <https://scikit-learn.org/>, [accessed 25-October-2021].
- [33] “Web de videojuegos,” <https://www.minijuegos.com/>, [accessed 10-November-2021].